University of Zurich[UZH]

Department of Informatics

Market Design for Cloud Computing and Software Markets

Dissertation submitted to the Faculty of Business,
Economics and Informatics
of the University of Zurich

to obtain the degree of
Doktor der Wissenschaften, Dr. sc.
(corresponds to Doctor of Science, PhD)

presented by
Ludwig Dierks
from Zurich, Switzerland

approved in February 2021

at the request of
Prof. Sven Seuken, Ph.D.
Prof. Ian A. Kash, Ph.D
Prof. Adam Wierman, Ph.D

The Faculty of Business, Economics and Informatics of the University of Zurich hereby authorizes the printing of this dissertation, without indicating an opinion of the views expressed in the work.

Zurich, February 17, 2021

The Chairman of the Doctoral Board: Prof. Dr. Thomas Fritz

Abstract

The digitalization of all parts of modern society makes life simpler and more convenient for many people. Yet, the underlying technological and economic systems become ever more complex. In many domains, this results in old or simplistic solutions being employed for new and complex problems. Consequently, much of the potential of digital markets remains untapped. In this thesis, I focus on two domains at the very center of any digital society: cloud computing markets and software markets. Using tools from microeconomics, operations research and computer science, I analyze these domains, identify inefficiencies and propose innovative market design solutions.

A first issue I focus on is that large amounts of capacity in cloud computing centers stand idle because they are reserved for purposes that do not actually require their continued use (e.g., for maintenance or for users with long-term contracts). As a potential solution to this problem, I focus on preemptible spot markets where users directly bid for capacity in a continuous auction. I model the cloud provider's profit optimization problem by combining queuing theory and game theory to analyze the equilibria of the resulting queuing system. I show that a provider can, under a mild condition, increase her profit over only offering a fixed-price market by also offering a spot market.

In a second research strand, I focus on the cluster admission control problem: Many modern cloud workloads are characterized by resource demands that change over time. Cloud resources are typically organized in compute clusters consisting of a few ten thousand compute cores and any request for more capacity of an existing workload has to be satisfied in the same cluster. A provider thus has to continuously decide whether she can add additional workloads to a given compute cluster. I formalize this problem as a constrained partially observable Markov decision process, which I then systematically relax to design fast heuristic admission policies. I fit the cluster admission model to a real world data trace and through simulations show substantial performance gains for the new policies compared to industry standards. I show how better prior information can further increase these gains and propose the use of variance-based pricing to improve priors. Using a duopoly market model, I analyze the competitive effects of such a pricing rule and show that employing variance-based pricing constitutes a competitive advantage.

Lastly, I focus on consumer software markets and study whether offering subscription licenses for a single product in addition or as an alternative to perpetual licenses can be used to increase a provider's revenue. I present a game-theoretic model where each user faces a simple Markov Decision Process when deciding whether he buys or subscribes. I derive the user equilibrium strategies and state the publisher's optimal pricing strategies. I show through numerical evaluations in the sub-domain of video games that it is typically, but not always, in a publisher's interest to offer subscription licenses alongside perpetual licenses. I further show that a publisher can often obtain a mild revenue increase by offering subscription licenses even without increasing the price of perpetual licenses.

Acknowledgements

Some advisors do not care for their students' research and some advisors care too much, simply using their students as help to write down their own ideas. Not Sven Seuken. Sven found the sweet spot where he fosters his students' own research agenda and leaves them the freedom to work on their own ideas, while still guiding them towards producing meaningful research. Thank you Sven, for your guidance and for your patience. Without it, I would not be where I am today.

I thank Ian Kash, for giving me the great opportunity of working with him on the cluster admission problem, for his support in finding an internship at Microsoft, for serving as second reader to my PhD proposal, and finally, for serving as external reviewer on this thesis. Similarly, I want to thank Adam Wierman for serving as the second external reviewer on this thesis. I am honored to have both of them on my dissertation committee.

I would like to thank Jacob LaRiviere for having me as an intern in the Microsoft Office of the Chief Economist and for giving me the opportunity to work on an interesting project with real data. I would also like to thank Ishai Menache, Aadharsh Kannan and Thomas Moscibroda for joining in and lending their varied expertise to this project. I would also like to thank the advisor of my Master's thesis, Felix Brandt, for his support when applying to this Ph.D. position.

I would like to thank all my former and current colleagues from the Economics and Computation Research Group at the University of Zurich for the many fruitful discussions we have had over the years: Timo Mennle, Gianluca Brero, Dmitrii Moor, Jakob Weissteiner, Mike Shann, Stefania Ionescu, Behnoosh Zamanlooy, Ermis Soumalias and Paul Friedrich. I would especially like to thank Steffen Schuldenzucker for always being available to read my drafts, Vitor Bosshard for his invaluable help with getting the ScienceCloud cluster to work with a Windows laptop and Nils Olberg for the opportunity to mentor him during the adoption market project.

I gratefully acknowledge financial support from Microsoft Research through its PhD Scholarship Programme.

I thank my family for their unyielding support. I thank my parents Jörn and Bärbel Dierks for the countless years of support; this would not have been possible without it. I thank Hanna Dierks for being a great sister and I thank my grandparents Hannelore and Herrmann Dierks for their endless love and understanding. I thank my late grandmother Hannelore Bräuer for supporting me going to Zurich, even though she felt certain not to witness my graduation. I miss you dearly. I thank my godfather Andreas Burkert for inspiring me to study math in the first place.

And lastly I want to thank my wife Chen-Ju Hung for her love and being so accepting of all the long weeks we had to be apart during my studies.

Contents

1 Motivation and Overview of Results[1]

1.1 Market Design for Digital Markets

The idea of trading goods and services is at least as old as human civilization, and, having been observed among animals (Noë and Hammerstein, 1995), likely predates humankind altogether. And with trade came marketplaces that each followed their own implicit and explicit rules. Over the ages, as human civilization evolved and production chains became ever more involved, so did markets (McMillan, 2003). One constant through the ages is that while new markets teem with potential, what worked before might not be optimal anymore. Those market participants that do not evolve with the markets are soon outcompeted and left in the dust.

Today, markets evolve faster than ever. The continuously increasing digitalization of all parts of modern society not only makes life simpler and more convenient for many people, it also causes the underlying systems to become ever more complex, both from a technological and an economic point of view. In many practical domains, this results in old or simplistic solutions still being applied to new and complex problems. Consequently, much of the potential of digital markets and systems remains untapped. In this thesis, I employ techniques from *game theory*, *artificial intelligence*, *data science* and *operations research* to develop new solutions to tap into some of this potential for two different domains at the center of any digitalized society: cloud computing and software markets.

1.1.1 Cloud Computing

Computers, smartphones, and apps are an integral part of modern life. Over the years, they have gotten ever more prevalent, their uses ever more sophisticated, and the devices themselves ever more portable. A development that, despite improvements in hardware, would not have been possible without the proliferation of cloud computing. The basic idea of cloud computing is that, instead of using local hardware, most computation should be run remotely on specialized server hardware. This not only means that consumers have to worry less about the computing power of their devices, but also allows companies and researchers to move their workloads out of local servers and into more efficient data centers, saving 20% to 50% of their IT costs in the process (Wauters et al., 2016). Using

[1] I liberally borrow from my own prior work (Dierks and Seuken, 2020*a*, Dierks, Kash and Seuken, 2019, Dierks and Seuken, 2020*b*,*c*) for parts of this chapter.

cloud resources further has manifold advantages beyond cost savings, such as on the spot scalability and flexibility, easy remote access and improved system security.

The low cost of cloud computing resources for customers is the result of a highly competitive market where cloud providers like Amazon EC2, Microsoft Azure and Google Cloud constantly strife to improve their offers. Nonetheless, most cloud clusters currently run at very low average utilization. This is problematic as a large part of the overall costs of running a cloud computing center is paid upfront when hardware is supplied and is therefore independent of usage. This low utilization is caused by various factors, such as technical limitations (i.e., the need to reserve capacity for node failures, maintenance etc.), outside factors (eg., fluctuations in overall demand), or inefficiencies in scheduling procedures, especially if virtual machines (VMs) might change size or do not use all of their requested capacity (Yan et al., 2016). Another cause is the nature of many modern workloads: highly connected tasks running on different VMs should be run on one cluster to minimize latency and bandwidth use (Cortez et al., 2017), making it important to keep enough free capacity in a cluster through the use of some kind of admission controller. While some of these technical factors, for example scheduling inefficiencies, are very well researched and understood, for others research is still at very early stages. Particularly lacking is research that merges the technological particularities of the domain with economic considerations and studies how particular market structures could be employed to increase the utilization. In this thesis, I apply market design techniques to increase the utilization and profit of cloud computing systems by either reselling idle capacity or reducing the need for idle capacity altogether.

1.1.2 Consumer Software Markets

Consumer software is an indispensable part of many peoples' lives. Be it productivity software (e.g., word processors or digital assistants), education software (e.g., language training programs), information software (e.g., maps or news), video games or other multi-media applications, almost every person utilizes some form of software in their daily lives.

Being only a few decades old, the markets for consumer software nonetheless already went through a large transformation from physical media like CDs and DVDs to purely digital forms of distribution. This transformation, combined with software markets growing into new domains like mobile phones, gave rise to many new business models and monetization schemes like in-app advertisement (Burns, Roseboom and Ross, 2016), microtransactions (i.e., the sale of many mini-upgrades for small amounts of money) or lootboxes (i.e., randomized microtransactions (Chen et al., 2020)). This rich domain has not seen much attention from market designers yet and the strategic implications of the different monetization schemes are not yet well understood. In this thesis, I embark on a foray into this domain by formally analyzing the revenue potential of subscription licenses

as an alternative or addition to traditional perpetual licenses for software products.

1.2 Background, Problem Statements and Research Questions

In this section, I give an overview of the background for the two domains I work on in this thesis and formulate my specific research questions.

1.2.1 Reducing Idle Capacity in Cloud Computing

The overarching goal of my work on cloud computing is to increase the utilization of cloud computing centers in ways that also increases the provider's profit. I have approached this from two different, yet complementary, directions: reselling capacity that is necessarily idle for its primary purpose on a secondary market and reducing the need for idle capacity altogether.

Reselling Idle Capacity

Cloud providers must constantly run many idle instances (virtual machines, compute containers, etc.), for example to guarantee service level agreements of long-term contracts, for maintenance, as fail safe redundancy, or simply as a buffer for future growth (Yan et al., 2016). While further advances in technology might in the future be able to reduce the need for this idle capacity, they cannot eliminate it. Instead, it might be more prudent to rent out the capacity to users as long as it is idle, but take it back whenever it is required for its primary purpose. As this capacity might be required for its primary purpose at a moment's notice, it cannot be sold on a standard cloud computing market. In today's cloud computing markets, instances are most commonly rented out via fixed-price offerings where users pay a fixed price per time unit and the provider aims to offer enough instances to be able to almost instantly satisfy all user requests. This approach is simple and reliable, satisfies the requirements of most users and is therefore widely used in practice. But as users in these markets expect to be able to use assigned resources as long as they want, idle capacity that might have to be taken back at a moments notice cannot be sold on there. Instead, a natural approach is to sell idle compute instances on a preemptible secondary market where users directly bid for capacity in a continuous auction and that is offered in addition to a cloud provider's primary fixed-price market. In such a market, any user that gets outbid is preempted and has to wait until the market price drops below his bid again. While this would clearly improve a cloud computing center's utilization, whether it would be in a provider's overall interest is not immediately clear.

Research Question 1 When can a provider utilize a spot market of idle capacity to increase her profit?

Reducing the Need for Idle Capacity

While there are many reasons for capacity to be idle, I focus on one that is relatively new and therefore has not yet gotten much attention: the nature of modern workloads. Today, workloads consist of ever more highly connected tasks running on different virtual machines (VMs) that need to communicate almost constantly and should be run on one physical compute cluster to minimize latency and bandwidth use (Cortez et al., 2017). Most cloud providers therefore offer this kind of service by bundling different VMs of a user together into a *deployment* of interdependent workloads. When the workload of a deployment changes, it can request a *scale out* in the form of additional VMs or shut some of its active VMs down. Scale out requests should almost always be accepted on the same cluster, as denying them would impair the quality of the service, possibly alienating customers. Providers consequently have to hold some parts of any cluster as idle reserves to guarantee that only a very low percentage of these requests is denied. Current practice for this typically just sees a fixed percentage of the cluster reserved. This may seem reasonable at first glance, as the law of large numbers might seem to suggest that with many jobs in a large cluster the current utilization would be a good guide to future utilization. But as Cortez et al. (2017) have shown, a relatively small number of deployments account for most of the utilization. This suggests that the *types of deployments* (i.e., small/large, fast/slow scaling, short/long lived etc.) currently in a cluster have a larger impact on the failure probability than is apparent, and policies that only take the current utilization into account are suboptimal: vast quantities of resources are reserved in anticipation of scale outs that will probably never happen. The amount of reserved capacity could be greatly reduced if information about deployments' future behavior would be taken into account to predict how many scale outs will occur.

Research Question 2 How can cluster admission policies be improved to take the scaling behavior of active deployments into account?

As information about deployment behavior might not be readily available, this research question also includes a learning and elicitation aspect. The elicitation approach I propose requires that the price a user pays for resources is partially based on the long-term variance of his deployments. Using such a pricing rule raises a broadly connected follow-up question about its general economic value that is distinct from its ability to help elicit information. In cloud computing and a number of other domains (e.g., mobile data or electricity markets), providers' costs are largely driven by how much buffer capacity they must keep in reserve. The providers' provisioning costs thus depend on how variable each user's demand is. Employing a variance-based pricing rule to (approximately) pass these

costs on to user seems natural. It has the advantage that low-variance users pay lower prices and are thus impacted less by the buffer requirements (which are mainly caused by high-variance users). This is not only fairer, but importantly, it incentivizes users to reduce their variance, which in turn reduces the provider's costs. Thus, in a monopolistic setting, the provider can obviously use variance-based pricing to increase his profits over what can be achieved with the industry standard of fixed per-unit prices. However, in a competitive market environment, the effects are less clear. The competitive pricing pressure by other providers may limit what a provider can achieve with variance-based pricing or lead to bad equilibria.

Research Question 3 Is variance-based pricing viable in competitive settings?

1.2.2 Subscription Licenses in Consumer Software Markets

While there are many emerging business models in software markets, in this thesis I focus on the revenue potential of one in particular: subscription licenses. Whereas a classic perpetual license, once bought, allows a user access to the product for as long as he desires (or, in some more recent cases, as long as the publisher supports it), a subscription license only allows access to the product for as long as the user pays a (typically monthly) recurring fee.[2] While in recent years, subscription licenses have become common for cloud-based Software-as-a-Service offerings (where their main selling point is access to cloud hardware), most products that do not come with significant cloud hardware are still only sold through perpetual licenses (though some publishers have recently experimented with subscription models (PC Gamer, 2020*a*,*b*)). In general, it is still unclear whether subscription licenses for consumer software could be used to noticeably increase revenue or whether the cannibalization of the market for perpetual licenses makes offering them suboptimal.

Research Question 4 Can a software publisher increase his revenue by offering subscription licenses alongside, or instead of, perpetual licenses.

1.3 Related Work

While the research questions in this thesis are all broadly situated in the field of market design (Roth, 2008), they fall into two methodological categories. Most of my research

[2] This is distinct from subscription services that give access to constantly changing bundles of products (e.g. Xbox Game Pass).

questions are microeconomic in nature and directly concern pricing rules as a way to increase profits (research questions 1, 3 and 4), while one research question focuses on technical considerations and utilizes pricing only as a supportive tool to enable information elicitation (research question 2).

Studying the viability of pricing rules is fundamentally a question of revenue management (Gallego and Van Ryzin, 1994, Chen, Farias and Trichakis, 2019, Schlosser et al., 2018). More specifically, one provider or publisher offering two slightly different versions of the same product to all users (i.e., spot and fixed-price instances or perpetual and subscription licenses) constitutes a form of product differentiation (Shaked and Sutton, 1982, Maskin and Riley, 1984, Desai, 2001). For example for posted price mechanisms with monotonically increasing marginal costs, it is known that some degree of product differentiation is typically profit optimal (Moorthy, 1984, Mussa and Rosen, 1978), though it is not directly clear how these results translate to the more complex types of product differentiation I analyze. For spot markets in cloud computing, Abhishek, Kash and Key (2017) have conversely shown that offering a spot market often decreases a provider's revenue, though they do not make any statement about profits. Separately, previous authors (e.g., Subramanya, Rizk and Irwin (2016)) have argued that spot markets will become unattractive once they become congested, as preemptions might become commonplace and destroy the value proposition of spot instances. The revenue effects of subscriptions have been studied for some other domains like ancillary services of a repeatedly sold core product (e.g., additional baggage for airline tickets) (Wang, Dada and Sahin, 2019) or professional Software-as-a-Service offerings (where, importantly, subscriptions provide scalable hardware while buy options do not, and utilities take a very different form than for consumer software) (Rohitratana and Altmann, 2012). Chawla et al. (2016) effectively studied a kind of subscription service with a free trial period for software products, but they restricted themselves to a single user type and evaluated their mechanism compared to extracting all expected value from the user.

Contrasting product differentiation stands price discrimination (Varian, 1989, Blattberg and Wisniewski, 1989, Gallego et al., 2006), where different users get offered the same product at different prices. Variance-based pricing lies at the intersection of these two concepts: while all users seemingly get offered the same product at different prices, the differences in provisioning costs caused by different user behavior mean that, from the provider's perspective, each user can be seen as obtaining a different product. Other approaches for dealing with varying demand include dynamic pricing and congestion-based pricing (Muratori and Rizzoni, 2015, Rong, Qin and An, 2018, Truong-Huu and Tham, 2014). These approaches focus on flattening demand peaks. A big downside is that these pricing rules make it unpredictable for users whether they can obtain the product at a given price when they need it. This puts providers who serve risk-averse users or users with relatively uniform but inelastic demand at a competitive disadvantage.

In difference to the other research questions, improving cluster admission policies is not primarily a question of pricing or revenue management, but of scheduling and matching users to hardware (Ashlagi et al., 2019, Ma and Simchi-Levi, 2019, Behnezhad and Reyhani, 2018, Assadi, Khanna and Li, 2017). While a lot of research has been done on scheduling *inside* the cluster (Schwarzkopf et al., 2013, Verma et al., 2015, Tumanov et al., 2016, Zhao et al., 2016) or on how multidimensional resources can be fairly divided among deployments (Ghodsi et al., 2011, Hindman et al., 2011), the admission problem has not been well studied before.

1.4 Publications Contained in This Thesis

This thesis consists of four papers, each addressing one of the research questions. This section restates the research questions and provides a list of the papers that address the respective research question.

Research Question 1 When can a provider utilize a spot market of idle capacity to increase her profit?

Publications

- **Cloud Pricing: The Spot Market Strikes Back**, Ludwig Dierks and Sven Seuken, Forthcoming in *Management Science*; previously appeared as an extended abstract in the *Proceedings of the 20th ACM Conference on Economics and Computation*, 2019.

Research Question 2 How can cluster admission policies be improved to take the scaling behavior of active deployments into account?

Publications

- **On the cluster admission problem for cloud computing**, Ludwig Dierks, Ian Kash and Sven Seuken, Working Paper, 2020; previously appeared as a 6-page extended abstract in the *Proceedings of the 14th Workshop on the Economics of Networks* (NetEcon'19), 2019.

Research Question 3 Is variance-based pricing viable in competitive settings?

Publications

- **The Competitive Effects of Variance-based Pricing**, Ludwig Dierks and Sven Seuken, Working Paper, 2020; an early version previously appeared in the *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence* (IJCAI'20), 2020.

Research Question 4 Can a software publisher increase his revenue by offering subscription licenses alongside, or instead of, perpetual licenses.

Publications

- **Revenue Maximization for Consumer Software: Subscription or Perpetual License?**, Ludwig Dierks and Sven Seuken, Working Paper, 2020.

1.5 Summary of Contributions

This section provides a brief summary of all four papers and explains how they answer the four research questions.

1.5.1 Cloud Pricing: The Spot Market Strikes Back

The first paper (Chapter 2) treats the first research question and provides a mild and easy-to-check condition under which a provider can increase her profit by offering a spot market.

To arrive at this result, I combine methods from queuing theory and game theory to analyze the equilibrium behavior of the users. I model the spot and fixed-price markets as distinct *queues* that arriving users can choose from, a modeling framework well studied for classical service systems (Hassin and Haviv, 2003, Hassin, 2016, Banerjee, Riquelme and Johari, 2015) and previously applied to cloud computing (Abhishek, Kash and Key, 2012, 2017, Gao, Iyer and Topaloglu, 2019).

In contrast to prior work, I specifically model all costs of the provider and the users that are required to perform a profit analysis. In my model, I assume that the provider incurs *fixed costs* for each instance in the fixed-price market and therefore only offers a finite number of fixed-price instances. For the spot market, I assume that the provider has a finite number of instances that she can offer without incurring any fixed costs (since she uses existing idle instances). For my main results, I assume that the provider takes those spot instances from a pool of idle resources that is distinct from the fixed-price instances (e.g., she instead takes them from long-term reserved instances, maintenance capacity, etc.), though I also show how the results translate to providers who want to use idle *fixed-price instances* (instead of some other, distinct pool of idle instances) for their spot market. For any instance (in the fixed-price or spot market), I assume that the provider incurs *load-dependent costs* whenever a job is running on it. Furthermore, I assume that a user has costs for waiting until his job is completed. Thus, his payoff consists of his value for job completion minus his incurred waiting cost and payment. Importantly, the model also includes *preemption costs* that a user incurs when his job is preempted. This takes the form of additional time required for the user's computation to again reach the point at which his job was preempted.

On the way towards answering the first research question, I first provide a full characterization of the user equilibria, i.e., I show how users will react depending on the provider's strategy. Any user equilibrium can be classified as one of four distinct types. In addition to equilibria where all users join the fixed-price market or all users join the spot market, there are two types of equilibria where users join both types of markets. These "hybrid equilibria" differ in whether the spot market is inherently faster or slower than the fixed-price market for users who submit a very high bid. I then give a characterization for which equilibrium a given provider strategy will result in based on how congested a "pure" spot market (i.e., without offering a fixed-price market) would be in relation to the chosen price in the fixed-price market.

With this equilibrium analysis in hand, I then derive a lower bound on the fixed-costs a provider saves when users move from the fixed-price to the spot market. Additionally, I upper bound the average time jobs require to run in the spot market, including time lost through preemptions. This bound makes use of the fact that, perhaps counter to intuition, users in the spot market can on average not be preempted more than once by another user. Given these two bounds, I then derive the central condition under which any provider who only offers a fixed-price market can increase her profit by also offering some of her idle instances on a spot market. Additionally, I show that a profit increase can always be combined with a Pareto improvement for the users. Informally, this condition requires that the idle capacity used for the spot market has to be sufficiently reliably idle and individual preemptions have to be sufficiently inexpensive compared to the ratio between fixed and load-dependent costs. Finally I illustrate the theoretical results through numerical examples that show the potential of sizable profit gains when offering spot markets.

In practice, these results can provide managerial guidance for cloud providers, in particular because the condition is easy to check and does not contain any queuing theoretic formulas. It is also mild enough to be satisfied for most cloud providers.

1.5.2 On the cluster admission problem for cloud computing

The second paper (Chapter 3) treats the second research question and provides new heuristic policies that cloud providers can employ to increase cluster utilization.

To derive these policies, I formalize the cluster admission problem as a constrained Partially Observable Markov Decision Process (POMDP) (Smallwood and Sondik, 1973) where each deployment behaves according to some stochastic process and the cluster tries to maximize the number of active compute cores without exceeding its capacity. Since the exact stochastic processes of individual arriving deployments are not known to the cluster, it has to reason about the observed behavior. Because optimally solving this POMDP is not feasible, I next propose a strategy for constructing heuristic policies via a series of carefully chosen simplifying assumptions. These assumptions reduce the highly

branching look-ahead space for the POMDP down to the approximation of a random variable using its moments. I then present the currently used threshold policy that does not take probabilistic information into account as well as two new policies. These new policies continuously refine their belief about the types of deployments currently in the cluster. When deciding whether a new deployment can be admitted, these policies utilize tail bounds to take successively higher moments of each deployment's stochastic processes into account. While the first policy approximates the risk of running over capacity using only the first moments (i.e., the expected values) of the processes with Markov's inequality, the second policy also takes the second moments (i.e., the variances) into account and to that end employs Cantelli's inequality, a one-sided refinement of Chebyshev's inequality.

To evaluate the impact of these policies, I fit a model to data from a real-world cloud computing center (Microsoft Azure internal jobs (Cortez et al., 2017)). Via simulations, I show that the higher moment policies produce a 30% improvement over current practice, which would translate to hundreds of millions of dollars a year in savings for large cloud providers.

As relatively little is known about arriving deployments, the observed performance gains from the more sophisticated policies are mainly driven by taking information about the current state of the cluster into account. I next examine how the utilization of the cluster can be further increased if more precise *prior information* about arriving deployments is available. To study the value of prior information, I introduce a simple framework to express the quality of information available in the form of samples from the true distributions underlying the deployment's stochastic processes. Through additional simulations, I quantify how the policies benefit from this additional information. Depending on the quality of information available, I find that the resulting gains increase to $50\% - 65\%$ relative to current practice.

Finally, given the importance of the quality of this information, I design a new information elicitation mechanism. The goal of this mechanism is to improve the cluster's utilization without requiring undue sophistication from the users. Users in cloud computing value simple and predictable pricing schemes. Requiring them to carry the risk of accidental misclassification or "unlikely" realizations goes against the very idea of simple cloud markets. Additionally, many users do not fully know the behavioral distributions of their own deployments. This makes using classic elicitation schemes such as scoring rules (Gneiting and Raftery, 2007) that are based on the report of behavioral distributions infeasible. Instead of explicitly asking users to describe the behavior of their deployments, I propose that cloud providers provide them with the opportunity to group their deployments into user-defined categories, each consisting of deployments with broadly similar characteristics. The cloud provider then takes on the task of estimating the behavior of each category for his admission policy and sets a small portion of the fee

for a deployment based on the variance of resource demands of deployments in a category. I show that such variance-based pricing provides users with the right incentives to (a) label their deployments properly (into, e.g., high and low variance deployments) and (b) structure their workloads in a way that helps the cluster run more efficiently. I lastly provide some additional simulations to quantify the benefits of an accurate labeling.

1.5.3 The Competitive Effects of Variance-based Pricing

The third paper (Chapter 4) treats the third research question and shows that introducing variance-based pricing typically constitutes a competitive advantage, even when no elicitation effects are taken into account.

To show this, I analyze a duopoly of providers who compete for a continuum of user types with stochastic demand. An important particularity of the domains I study for this research question is that each provider has to always provision enough resources to satisfy the total demand of his users with high probability. To keep the model concise, this is abstractly modeled through cost functions that are monotonically increasing in the demand variance of the respective provider's users. For domains with roughly constant long-term user bases and providers that are interested in their long-term profit, this is a reasonable abstraction. I assume that each provider either conservatively employs constant per-unit prices or is willing to innovate and employ per-unit prices that linearly depend on each user's variance. I only consider linear prices because their simplicity makes them most plausible and marketable in practice.

I first analyze the Bayes-Nash equilibria when both providers are conservative and only charge constant prices. In this case, the problem becomes similar to a classic Bertrand competition and the provider that is more cost efficient for the whole set of user types can extract the cost difference to the second provider as profit. I then show that, as long as the cost functions of both providers are reasonably close, unilaterally switching to variance-based pricing can be used to obtain a higher profit. While no Bayes-Nash equilibria exist in this case, as one of the two providers always has an advantageous deviation, the innovating provider can typically "force" the conservative provider to react in a way that results in higher profit for the innovator than any constant equilibrium. One way to think about this is through the lens of Stackelberg equilibria, i.e., an equilibrium concept where one provider, the leader, first commits to a strategy and the other provider, the follower, then reacts with his own strategy. Under some mild cost conditions and assuming the other provider only plays constant strategies, I show that there always exists a Stackelberg equilibrium where the leader innovates to variance-based prices and obtains higher profit than he could obtain when playing constant pricing strategies. If the other provider is willing to also employ variance-based pricing, I show that, as long as splitting a provider's population of users between symmetric copies of the same provider would strictly increase costs, well chosen variance-based prices will not lead to less profit

than constant prices. But as an innovative provider now loses much of the power his larger strategy space gave him against a purely conservative provider, a strict profit increase can no longer be guaranteed. For example, the provider cannot achieve non-zero profit when both providers have the same cost function.

I then characterize all Bayes-Nash equilibria that arise if both providers employ variance-based pricing. As long as providers are not symmetric in their cost functions, the profits of both providers here often increase, as they can specifically attract user types that they can better serve due to their respective cost functions. Finally, I show that the social welfare of the market may decrease if only one provider employs variance-based pricing, but that it at least weakly increases over the constant price equilibria if both employ variance-based pricing. Overall, this work shows that variance-based pricing is a viable approach in most competitive settings and I recommend providers to further explore it as an option for their markets.

1.5.4 Revenue Maximization for Consumer Software: Subscription or Perpetual License?

The fourth paper (Chapter 5) treats the fourth research question and shows that offering subscription licenses alongside classical perpetual licenses can typically be used to increase a software publisher's revenue.

There are a few important particularities of the domain that have to be taken into account when trying to show the revenue potential of subscription licenses. Software as a purely digital good neither has a limited stock nor marginal costs. Instead, the quality of the product in the eyes of users continuously decays (Mao et al., 2018) and any user might separately lose interest at some point. Additionally, optional paid upgrades[3] are often employed to counteract this relative quality decay and rekindle lost interest.

To properly take account of these particularities, I introduce a tailor-made model that takes the form of a two-step game. In the first step, the publisher chooses his pricing strategy, setting the prices of perpetual licenses for his product and one upgrade, as well as the per-timestep price of subscription licenses. Given these prices, the users then act inside a discrete time sub-game where they dynamically arrive over time. While each user arrives with demand for the product, he has a certain chance of losing this demand over time. Losing demand sets the user's future utility for having access to the product to zero. Once a user has lost demand, only the release of a new upgrade has a chance to rekindle his interest. Additionally, the value of the product in the eyes of each user decays with a user-dependent rate over time. For this sub-game, I show that there are only five distinct classes of user equilibrium strategies, which significantly aids in our analysis. Based on this, I derive the publisher's revenue as a function of his pricing strategy and show that only offering a subscription option can never be optimal, though

[3] In the domain of video games, such upgrades are typically called "downloadable content" (DLC).

it can be far better than only offering perpetual licenses.

To get an impression of the impact we can expect in practice, I then turn to a data trace consisting of price, ownership and activity data from the domain of video games. This data was obtained from the website *Steam Spy* [4], which collects publicly available data from the large video game storefront *Steam*[5] and uses it to statistically estimate the number of owners of video games over time and what percentage of them actively used the game recently (i.e., in the last two weeks). As this data does not contain any information about user preferences or users that did not buy the product, it unfortunately does not allow us to directly validate our model. As even the available information is somewhat unreliable, I chose not to directly fit the model to the data for any single game. I instead use the dataset to inform a reasonable generic parametrization. I then numerically compare offering the different license types for this parametrization and find that while subscription licenses slightly outperform perpetual licenses, but are in turn significantly outperformed by offering both types of licenses.

Performing comparative statics starting from this generic parametrization, I then show that as long as there is sufficient heterogeneity in the expected time until users lose interest, offering both types of licenses can lead to revenue increases between 10% and 40% for the provider. Importantly though, this is not mainly achieved by attracting additional users. Instead, the additional revenue is to a large part the result of better utility extraction from users through product discrimination. When offering both types of licenses, the provider can charge a higher buy price from long-term users, because users that only expect to use the product for a short time can switch to subscribing. If no subscription licenses are offered, such higher prices instead would lead to short-term users not obtaining the product at all. Typically, the social welfare of the system consequently slightly decreases when offering both licenses with revenue optimal prices. That said, if the publisher desires, I show that a modest revenue increase over only offering perpetual licenses can typically be achieved without decreasing the utility of any user.

1.6 Conclusion and Future Work

Modern digital markets all have distinct traits and interactions with the underlying technological systems that often make classic economic approaches suboptimal. Instead, a rigorous analysis of these markets is required to find tailored strategies that can realize their full potential. In this thesis, I have analyzed two modern digital markets: cloud computing and software markets. I have shown some of the inefficiencies inherent to current practices and proposed new approaches to alleviate them. For cloud computing, I have shown that offering a spot market of idle capacity can be used to increase a cloud provider's profit and that adaptive cluster admission policies, aided by variance-

[4] https://steamspy.com/about
[5] https://store.steampowered.com/about/

based pricing rules, can greatly increase the utilization of a cloud computing center. Additionally, I have shown that variance-based pricing rules are a viable strategy in competitive settings with long-term provisioning costs such as cloud computing markets. Lastly, for software markets, I have shown that software publishers can typically utilize subscription licenses to greatly increase their revenue.

Future Work

I see a number of promising future research threads in both domains. First, in cloud computing, it would be interesting to compare dynamically priced secondary spot markets to alternatively offering a secondary fixed-price market. Such a secondary fixed-price market would offer instances at a lower price than the primary market and also include preemptions when the instance is required for it's primary purpose. While this would produce less profit than a spot market when preemptions are completely lossless, the higher rate of preemptions in spot markets might mean that it is still desirable in practice. It would be very interesting to derive a condition for when either type of market produces higher profit, though the required queuing theoretic formulas are highly complex. Further simplifying assumptions compared to the model I employed to analyze spot markets in this thesis might therefore be required.

A second interesting question in cloud computing would be to extend the analysis of the cluster admission problem to the more general cluster assignment problem, i.e., the question to which of multiple clusters an arriving deployment should be assigned. The work included in this thesis implicitly assumes that each cluster employs the same admission policy and gets sent deployments from the same population mix of deployments. While this is consistent with industry practice, optimizing over multiple clusters at the same time might yield even better results. As finding optimal cluster admission policies is already infeasible, there is little hope to optimally solve the cluster assignment problem. But this does not preclude further advances over the status quo. In an ongoing project with Jacob LaRiviere, Ishai Menache, Aadharsh Kannan and Thomas Moscibroda from Microsoft I am already investigating whether dedicating different clusters to different sub-populations of deployments could improve the achievable utilization. Even with simple threshold policies, we find gains of up to 10% over only employing a single type of cluster.

For software markets, there are also a number of interesting future research directions. While I have looked at a first monetization scheme for consumer software markets in this thesis, this barely scratches the surface of this domain. Among the many monetization schemes that recently appeared in practice, few have gotten much attention from market designers. One particularly interesting direction would be to analyze the incentive structure and revenue implications of bundled subscription services such as Microsoft's Xbox Game Pass or Sony's Playstation Now. Similarly to music or video streaming

platforms, bundled software subscriptions allow users to pay a small monthly fee and obtain access to a constantly changing library of titles, often from multiple publishers. While the advantages for users are relatively apparent, it is less clear under what circumstances it is in a publisher's interest to add their product to such a service and what the resulting equilibria are. It would be especially interesting to see how the widespread adoption of subscription services changes the value proposition of products not sold on it, i.e., whether it would lead to an increase or decrease in equilibrium prices for these products.

Another interesting question in the domain of consumer software markets is the effect of microtransactions on the incentives for product design. Microtransactions effectively split a product into a menu of smaller items. Instead of having to buy the whole product, users can just buy the parts they value highly enough. For a given product, as long as users are sufficiently heterogeneous, this product differentiation can yield a stark revenue increase for the publisher. But it also influences the publishers' incentives when designing a new product, as design choices that enable certain types of microtransactions might reduce the appeal for other users compared to the optimal monolithic product. A monolithic product needs to pose a good value proposition to a relatively large part of the user base at the same time, as they all pay the same price. A menu-based product thus faces a quite different optimization problem. It would be interesting to formally analyze these differences in a game-theoretic model to see how they affect social welfare in equilibrium. As the equilibria are likely to vary greatly depending on whether a product faces competition in their specific niche or not, both a monopoly and a duopoly analysis would have merit.

Bibliography

Abhishek, Vineet, Ian A. Kash, and Peter Key. 2012. "Fixed and Market Pricing for Cloud Services." In *2012 Proceedings IEEE INFOCOM Workshops*.

Abhishek, Vineet, Ian A. Kash, and Peter Key. 2017. "Fixed and Market Pricing for Cloud Services." CoRR abs/1201.5621. Extended version of Abhishek et al. (2012).

Ashlagi, Itai, Maximilien Burq, Chinmoy Dutta, Patrick Jaillet, Amin Saberi, and Chris Sholley. 2019. "Edge weighted online windowed matching." In *Proceedings of the 2019 ACM Conference on Economics and Computation*.

Assadi, Sepehr, Sanjeev Khanna, and Yang Li. 2017. "The stochastic matching problem: Beating half with a non-adaptive algorithm." In *Proceedings of the 2017 ACM Conference on Economics and Computation*.

Banerjee, Siddhartha, Carlos Riquelme, and Ramesh Johari. 2015. "Pricing in Ride-share Platforms: A Queueing-Theoretic Approach." *Proceedings of the 16th ACM Conference on Economics and Computation*, 639.

Behnezhad, Soheil, and Nima Reyhani. 2018. "Almost optimal stochastic weighted matching with few queries." In *Proceedings of the 2018 ACM Conference on Economics and Computation*.

Blattberg, Robert C, and Kenneth J Wisniewski. 1989. "Price-induced patterns of competition." *Marketing Science*, 8(4): 291–309.

Burns, Zachary, Isaac Roseboom, and Nicholas Ross. 2016. "The Sensitivity of Retention to In-Game Advertisements: An Exploratory Analysis." In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Chawla, Shuchi, Nikhil R. Devanur, Anna R. Karlin, and Balasubranianian Sivan. 2016. "Simple Pricing Schemes for Consumers with Evolving Values." In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*. USA:Society for Industrial and Applied Mathematics.

Chen, Ningyuan, Adam N. Elmachtoub, Michael Hamilton, and Xiao Lei. 2020. "Loot Box Pricing and Design." In *Proceedings of the 21st ACM Conference on Economics and Computation*. New York, NY, USA:Association for Computing Machinery.

Chen, Yiwei, Vivek F Farias, and Nikolaos Trichakis. 2019. "On the efficacy of static prices for revenue management in the face of strategic customers." *Management Science*, 65(12): 5535–5555.

Cortez, Eli, Anand Bonde, Alexandre Muzio, Mark Russinovich, Marcus Fontoura, and Ricardo Bianchini. 2017. "Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms." In *Proceedings of the 26th Symposium on Operating Systems Principles*.

Desai, Preyas S. 2001. "Quality segmentation in spatial markets: When does cannibalization affect product line design?" *Marketing Science*, 20(3): 265–283.

Dierks, Ludwig, and Sven Seuken. 2020*a*. "Cloud pricing: The spot market strikes back." *Forthcoming in Management Science*.

Dierks, Ludwig, and Sven Seuken. 2020*b*. "The Competitive Effects of Variance-based Pricing." In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*.

Dierks, Ludwig, and Sven Seuken. 2020*c*. "Revenue Maximization for Consumer Software: Subscription or Perpetual License?" *Working Paper*.

Dierks, Ludwig, Ian Kash, and Sven Seuken. 2019. "On the cluster admission problem for cloud computing." *arXiv preprint arXiv:1804.07571*, `https://arxiv.org/abs/1804.07571`.

Gallego, Guillermo, and Garrett Van Ryzin. 1994. "Optimal dynamic pricing of inventories with stochastic demand over finite horizons." *Management science*, 40(8): 999–1020.

Gallego, Guillermo, Woonghee Tim Huh, Wanmo Kang, and Robert Phillips. 2006. "Price competition with the attraction demand model: Existence of unique equilibrium and its stability." *Manufacturing & Service Operations Management*, 8(4): 359–375.

Gao, Jiayang, Krishnamurthy Iyer, and Huseyin Topaloglu. 2019. "When fixed price meets priority auctions: Competing firms with different pricing and service rules." *Stochastic Systems*, 9(1): 47–80.

Ghodsi, Ali, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. 2011. "Dominant resource fairness: Fair allocation of multiple resource types." In *USENIX Symposium on Networked Systems Design and Implementation*.

Gneiting, Tilmann, and Adrian E Raftery. 2007. "Strictly proper scoring rules, prediction, and estimation." *Journal of the American statistical Association*, 102(477): 359–378.

Hassin, Refael. 2016. *Rational Queueing*. Boca Raton, FL:CRC Press.

Hassin, Refael, and Moshe Haviv. 2003. *To queue or not to queue: Equilibrium behavior in queueing systems*. Norwell, MA:Kluwer Academic Publishers.

Hindman, Benjamin, Andy Konwinski, Matei Zaharia, Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, and Ion Stoica. 2011. "Mesos: A Platform for Fine-grained Resource Sharing in the Data Center." In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*. Berkeley, CA, USA:USENIX Association.

Mao, Weichao, Zhenzhe Zheng, Fan Wu, and Guihai Chen. 2018. "Online Pricing for Revenue Maximization with Unknown Time Discounting Valuations." In *IJCAI*.

Maskin, Eric, and John Riley. 1984. "Monopoly with incomplete information." *The RAND Journal of Economics*, 15(2): 171–196.

Ma, Will, and David Simchi-Levi. 2019. "Tight Weight-dependent Competitive Ratios for Online Edge-weighted Bipartite Matching and Beyond." In *Proceedings of the 2019 ACM Conference on Economics and Computation*.

McMillan, John. 2003. *Reinventing the bazaar: A natural history of markets*. WW Norton & Company.

Moorthy, K Sridhar. 1984. "Market segmentation, self-selection, and product line design." *Marketing Science*, 3(4): 288–307.

Muratori, Matteo, and Giorgio Rizzoni. 2015. "Residential demand response: Dynamic energy management and time-varying electricity pricing." *IEEE Transactions on Power systems*, 31(2): 1108–1117.

Mussa, Michael, and Sherwin Rosen. 1978. "Monopoly and product quality." *Journal of Economic Theory*, 18(2): 301 – 317.

Noë, Ronald, and Peter Hammerstein. 1995. "Biological markets." *Trends in Ecology & Evolution*, 10(8): 336–339.

PC Gamer. 2020*a*. "Paradox is testing a subscription service for Europa Universalis 4." `https://www.pcgamer.com/paradox-is-testing-a-subscription-service-for-europa-universalis-4/`.

PC Gamer. 2020*b*. "Ubisoft clarifies that Trackmania is subscription-based." `https://www.pcgamer.com/ubisoft-says-trackmania-is-not-subscription-based-you-just-pay-for-it-multiple-times/`.

Rohitratana, Juthasit, and JöRn Altmann. 2012. "Impact of pricing schemes on a market for Software-as-a-Service and perpetual software." *Future Generation Computer Systems*, 28(8): 1328–1339.

Rong, Jiang, Tao Qin, and Bo An. 2018. "Dynamic Pricing for Reusable Resources in Competitive Market with Stochastic Demand." In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Roth, Alvin E. 2008. "What have we learned from market design?" *Innovations: Technology, Governance, Globalization*, 3(1): 119–147.

Schlosser, Rainer, Carsten Walther, Martin Boissier, and Matthias Uflacker. 2018. "Data-Driven Inventory Management and Dynamic Pricing Competition on Online Marketplaces." In *IJCAI*.

Schwarzkopf, Malte, Andy Konwinski, Michael Abd-El-Malek, and John Wilkes. 2013. "Omega: flexible, scalable schedulers for large compute clusters." In *Proceedings of the 8th ACM European Conference on Computer Systems*.

Shaked, Avner, and John Sutton. 1982. "Relaxing price competition through product differentiation." *The review of economic studies*, 3–13.

Smallwood, Richard D, and Edward J Sondik. 1973. "The optimal control of partially observable Markov processes over a finite horizon." *Operations research*, 21(5): 1071–1088.

Subramanya, Supreeth, Amr Rizk, and David Irwin. 2016. "Cloud Spot Markets are Not Sustainable: The Case for Transient Guarantees." In *8th USENIX Workshop on Hot Topics in Cloud Computing*.

Truong-Huu, Tram, and Chen-Khong Tham. 2014. "A novel model for competition and cooperation among cloud providers." *IEEE Transactions on Cloud Computing*, 2(3): 251–265.

Tumanov, Alexey, Timothy Zhu, Jun Woo Park, Michael A Kozuch, Mor Harchol-Balter, and Gregory R Ganger. 2016. "TetriSched: global rescheduling with adaptive plan-ahead in dynamic heterogeneous clusters." In *Proceedings of the Eleventh European Conference on Computer Systems*.

Varian, Hal R. 1989. "Price discrimination." *Handbook of industrial organization*, 1: 597–654.

Verma, Abhishek, Luis Pedrosa, Madhukar Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. 2015. "Large-scale cluster management at Google with Borg." In *Proceedings of the Tenth European Conference on Computer Systems*.

Wang, Ruxian, Maqbool Dada, and Ozge Sahin. 2019. "Pricing ancillary service subscriptions." *Management Science*, 65(10): 4712–4732.

Wauters, Patrick, Sebastiaan Van Der Peijl, Valentina Cilli, Marco Bolchi, Pawel Janowski, Marie Moeremans, Hans Graux, Graham Taylor, and Diana Cocoru. 2016. "Measuring the economic impact of cloud computing in Europe." Deloitte Study for the European Commision 133.

Yan, Ying, Yanjie Gao, Yang Chen, Zhongxin Guo, Bole Chen, and Thomas Moscibroda. 2016. "TR-Spark: Transient Computing for Big Data Analytics." In *Proceedings of the 7th ACM Symposium on Cloud Computing*.

Zhao, J., K. Yang, X. Wei, Y. Ding, L. Hu, and G. Xu. 2016. "A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment." *IEEE Transactions on Parallel and Distributed Systems*, 27(2): 305–316.

2 Cloud Pricing: The Spot Market Strikes Back

The content of this chapter has previously appeared in:

> Ludwig Dierks, Sven Seuken (2021) **Cloud Pricing: The Spot Market Strikes Back**. *Management Science*;
>
> Ludwig Dierks, Sven Seuken (2019) **Cloud Pricing: The Spot Market Strikes Back** (extended abstract). *Proceedings of the 20th ACM Conference on Economics and Computation.*

Cloud Pricing: The Spot Market Strikes Back*

September 2020

LUDWIG DIERKS, University of Zurich, Switzerland
SVEN SEUKEN, University of Zurich, Switzerland

Cloud computing providers must constantly hold many idle compute instances available (e.g., for maintenance, or for users with long-term contracts). A natural idea, which should intuitively increase the provider's profit, is to sell these idle instances on a secondary market, for example, via a preemptible spot market. However, this ignores possible "market cannibalization" effects that may occur in equilibrium as well as the additional costs the provider experiences due to preemptions. To study the viability of offering a spot market, we model the provider's profit optimization problem by combining queuing theory and game theory to analyze the equilibria of the resulting queuing system. Our main result is an easy-to-check condition under which a provider can simultaneously achieve a profit increase and create a Pareto improvement for the users by offering a spot market (using idle resources) alongside a fixed-price market. Finally, we illustrate our results numerically to demonstrate the effects the provider's costs and her strategy have on her profit.

1 INTRODUCTION

Providers of cloud services like Amazon EC2 or Microsoft Azure rent out computing capacity to millions of users. These cloud services generate billions of dollars in yearly revenue and the market for these services is still growing exponentially.[1] While some users enter into year-long contracts, many prefer to obtain resources on-demand (i.e., just when they need them and without long-term commitments). Today, on-demand compute resources (i.e., CPU, RAM, bandwidth, etc.) are most commonly combined into compute instances (e.g., virtual machines) and rented out via *fixed-price* markets. In these markets, users pay a fixed price per time unit and the provider aims to keep enough instances available to be able to almost instantly satisfy all requests. This approach is considered to be simple, reliable, and to satisfy the requirements of most users; it is therefore widely used in practice.[2] At the same time, all cloud computing centers always contain many *idle* instances: for example, to guarantee the service level agreements of long-term contracts, for maintenance, as fail-safe redundancy, or simply as a buffer for future growth (Yan et al. 2016). In effect, a sizable number of instances do nothing at any given time. However, low utilization rates are inherently undesirable, because most of the overall per-instance costs are independent of utilization (Barroso et al. 2018). A lot of research has been done to increase these utilization rates, e.g., by allowing advance reservation of resources (Azar et al. 2015, Babaioff et al. 2017), by predicting future usage (Cohen et al. 2019, Cortez et al. 2017, Dierks et al. 2019, Jyothi et al. 2016) or by incentivizing users to reduce the variance of their demand (Dierks and Seuken 2020). However, any cloud computing center will always have a non-negligible number of idle instances.

Naturally, a cloud provider may also want to sell those idle instances; but unfortunately, they are only "temporarily" idle. Whenever an instance is needed for its primary purpose (e.g., for maintenance or for a user with a long-term contract), then the job that is currently running on the instance must be *preempted*, i.e., the job is shut down. Of course, getting preempted and restarting a job on a different instance is not lossless for a user; for example, because a new instance may not be immediately available, because unsaved progress to completion may be lost, or because it may require some time to reconfigure the new instance. For this reason, idle instances cannot be

*This paper is accepted and forthcoming in Management Science. Some of the ideas presented in this paper were also described in a one-page abstract that was published in the conference proceedings of EC'19 (Dierks and Seuken 2019).

[1] https://www.microsoft.com/en-us/Investor/earnings/FY-2018-Q2/press-release-webcast

[2] For example: https://aws.amazon.com/ec2/pricing/on-demand/ or https://azure.microsoft.com/en-us/offers/pay-as-you-go/

sold on the regular fixed-price market where users are guaranteed the continued (non-preemtible) usage of their resources. However, researchers (e.g., Abhishek et al. (2017) and Hoy et al. (2016)) as well as cloud providers (e.g., *Amazon EC2 Spot instances* and *Microsoft Azure Low Priority VMs*)[3] have considered to instead sell the idle instances on a secondary (cheaper) market where the users know that their jobs may be preempted. A user would use this market with the understanding that, when his job is shut down, he can restart the job once another instance becomes available.[4]

Given that the supply of idle resources changes over time, "dynamic pricing" is a natural choice for the secondary market.[5] Since in any secondary market consisting of idle instances the users have to cope with preemptions by the nature of the instances, additional preemptions caused by dynamic pricing do not qualitatively change the user experience. We focus on dynamic pricing that is implemented as a *preemptible spot market*. This means that users bid for resources and are served whenever the current market price is lower than their bid. If the market price rises too far while their job is running, they are preempted until the price drops again. Note that the idea of using preemptible spot markets for unused resources is not new: similar spot markets have also recently been proposed for the power capacity in multi-tenant data centers (Islam et al. 2018).

At first sight, it may seem obvious that the provider should offer idle instances on such a secondary (spot) market — after all, these instances seem "free" for the provider and selling them (at essentially any price) generates revenue. However, this thinking ignores two important points. First, putting otherwise idle instances under load causes additional load-dependent costs for the provider, which can be much larger in a *preemptible* secondary market than in a *non-preemptible* market. Second, it ignores possible "market cannibalization" effects that may occur in equilibrium, i.e., that users may choose to move from the more expensive fixed-price market to the cheaper spot market. Indeed, Abhishek et al. (2017) have shown that certain cannibalization effects can occur, at least in terms of revenue. Intuitively, market cannibalization becomes particularly problematic when the preemption costs on the spot market are large (i.e., when the users get preempted often and need to re-run large parts of their jobs), because then most users are only willing to pay very little for joining the spot market.

In this paper, we ask the following research question: *when can a cloud provider offer a spot market in addition to a fixed-price market to increase her profit?* Note that a provider trying to maximize her profit faces two basic questions: (1) what *price* should she ask for instances in the fixed-price market, and (2) how many (possibly zero) spot instances should she offer? Her profit then depends on the users' actions, given the offered markets. To answer our research question, we combine queuing theory and game theory to analyze the equilibrium behavior of the users. We model the two different markets as distinct *queues* that arriving users can choose from, a modeling framework well studied for classical service systems (Banerjee et al. 2015, Hassin 2016, Hassin and Haviv 2003) and previously applied to cloud computing (Abhishek et al. 2012, 2017, Gao et al. 2019). While such a queueing-theoretic approach is not the only viable approach to model cloud markets (Hoy et al. 2016, Kash and Key 2016, Zhang et al. 2016), it is particularly well suited to model the temporal nature of users continuously arriving and departing.

In contrast to prior work, we present a significantly more realistic model for the cloud domain that also captures all relevant costs of the provider and the users (Section 3), enabling us to perform

[3]See https://aws.amazon.com/ec2/spot/ and https://azure.microsoft.com/en-us/pricing/details/batch/. Note that, Agmon Ben-Yehuda et al. (2013) collected evidence, suggesting that, at some point, the EC2 spot market (despite its name) may not actually have used real spot pricing.

[4]To distinguish the provider from the users, we use "she/her" when referring to the provider and "he/his" for a user.

[5]Note that there are alternative pricing mechanisms for the secondary market that are also plausible. For example, the secondary market could also use fixed prices and this would likely increase the provider's profit as well. Future work should compare the two alternatives; but before we can do so, we first need to analyze the spot market case.

a profit analysis. First, we assume that the provider incurs *fixed costs* for each instance in the fixed-price market and therefore only offers a finite number of fixed-price instances. For the spot market, we assume that the provider has a finite number of instances that she can offer without incurring any fixed costs (since she uses existing idle instances). We assume that the provider takes those spot instances from a pool of idle resources that is distinct from the fixed-price instances (e.g., long-term reserved instances, maintenance capacity, etc.). For most large cloud providers, a sizable pool of such instances exists (Yan et al. 2016), and these instances are more "reliably idle" than most instances on the fixed-price market.[6] We assume that, for any instance (in the fixed-price or spot market), the provider incurs *load-dependent costs* whenever a job is running on it. Furthermore, we assume that a user has costs for waiting until his job is completed. Thus, his payoff consists of his value for job completion minus his incurred waiting cost and payment. Importantly, our model also includes *preemption costs* that a user incurs when his job is preempted. This takes the form of additional time required for the user's computation to again reach the point at which his job was preempted.

On the way towards answering our research question, we first provide a full characterization of the resulting user equilibria depending on the provider's strategy (Section 4). Our main result is a condition (see Definition 5.3 in Section 5.2) under which any provider who only offers a fixed-price market can increase her profit and simultaneously create a Pareto improvement for the users by also offering some of her idle instances on a spot market (Sections 5.1 and 5.2).[7]

In practice, our results can provide managerial guidance for cloud providers, in particular because our condition is easy to check (Section 5.3). Our condition is also mild enough to be satisfied for most cloud providers. Furthermore, we discuss how a cloud provider may increase her profit even if she is unable to compute her optimal strategy.

To illustrate our theoretical results, we numerically calculate equilibria for multiple examples where users arrive according to a Poisson process and require exponentially distributed service times (Section 6). We use these examples to study the effect that different cost structures have on the profitability of offering a spot market, and how the provider's strategy impacts her profit. In particular, we illustrate our main result, i.e., how a profit increase can be combined with a Pareto improvement for the users. Further, we show that even under relatively pessimistic conditions for a spot market, sizeable profit increases can still be attainable.

2 RELATED WORK

Offering a spot market is a form of *product differentiation* (Desai 2001, Maskin and Riley 1984, Shaked and Sutton 1982), where a provider offers *differentiated* products to appeal to different users. This contrasts with standard *price discrimination*, where typically identical or very similar products are sold at varying prices (Varian 1989). In their seminal paper on product differentiation, (Mussa and Rosen 1978) first formalized the relationship between prices and a user's obtained quality level, similar to results (Myerson 1981) later obtained for optimal auctions. For posted price mechanisms, (Mussa and Rosen 1978) further showed that optimal pricing policies can result in the segmentation of users into some segments where all users obtain the same product (as in our fixed-price market), while for all other users (who do not balk), the obtained product quality increases monotonically in

[6] In Appendix D, we provide intuition for why the fixed-price market, even if it is large, might not have a lot of reliably idle instances. In D, we also show how our results translate to providers who nevertheless want to use idle *fixed-price instances* (instead of some other, distinct pool of idle instances) for their spot market.

[7] The ability to combine the profit increase with a Pareto improvement for the users is particularly important in practice, where the competition between providers like Google, Microsoft, or Amazon is fierce. Thus, any strategy for increasing a provider's profits must guarantee that users are not worse off than before to ensure that they do not migrate to competing providers.

their type (as in our spot market). (Moorthy 1984) later extended the model of (Mussa and Rosen 1978) to non-linear user preferences (for discrete user types), showing that limiting the number of offered quality levels is often profit-optimal. These classic product differentiation papers give hope that offering a spot market alongside a fixed-price market may be profit increasing in some cases. However, because their analysis only considers simple posted price mechanisms (where users have no *direct* effect on each other) it does not apply to our cloud computing setting where users' actions have more complicated effects. In our problem, the values of the users for "obtaining a product" (i.e., running a job in a particular market) are not fixed but depend on the user equilibrium. Furthermore, the costs of the provider for offering a product are also not fixed but again depend on the user equilibrium. This is why we need to combine queuing theory and game theory (see, e.g., Hassin and Haviv (2003)) to answer the question when offering a spot market increases profits. An example of combining these techniques for revenue management is the work by Afèche (2013), who used a single server queue to show that it is often revenue optimal to introduce artificial delays when selling a product to time sensitive customers.

The work most related to ours is (Abhishek et al. 2012, 2017), who like us study cloud computing markets. Interestingly, the authors found that offering a spot market often decreases the provider's revenue. However, this does not contradict our results: their model was tailored towards a revenue analysis and therefore could assume an infinite number of fixed-price instances and did not need to model the provider's costs. Given this, they could not make any statements about profits. Gao et al. (2019) used a similar modeling approach to study the competition between two firms, where one firm only offers a fixed-price market of fixed finite size (i.e., that is not necessarily large enough for demand) while the other firm only offers a spot market.

Recently, a series of papers has studied the problem of selling resources through more complicated auction and pricing mechanisms that take individual requirements of jobs into account, including deadlines (Zhou et al. 2017), multi-dimensional resource requirements (Shi et al. 2014, Zhang et al. 2014), and the provider's opportunity cost for scheduling a given job (Boodaghians et al. 2019, Kash et al. 2017). However, in contrast to our model, these papers do not consider the following important business constraint: in practice, any provider who wants to keep her market share *must* also offer a non-preemptible fixed-price market alongside any other offerings, because many users want access to a fixed-price market. Furthermore, most of the prior work on cloud spot markets (including the papers cited above) do not consider the users' preemption costs. However, preemption costs are an important factor to analyze the profitability of the spot market, and previous authors (e.g., Subramanya et al. (2016)) have even argued that spot markets may become too unattractive once they become congested. However, we show that the costs incurred due to preemption are bounded, such that even congested spot markets remain attractive for the provider.

3 MODEL

In this section, we introduce our model. Before starting with the formal definitions, we provide some brief intuition for our framework. To analyze the profit a cloud provider obtains from running the different markets, we need to consider how her decisions affect the actions of potential users. To this end, we define a two-step model that is reminiscent of a Stackelberg game (Maharjan et al. 2013) with an important difference. As in a Stackelberg game, in a first step, the cloud provider chooses her actions (i.e., what markets to offer). This defines the parameters of the game the users will play in the next step. In the second step, with the provider's strategy fixed, the users then play the resulting game only with each other; i.e., they decide which market to join and potentially what to bid. In contrast to a Stackelberg game, the sub-game in the second step takes the form of a queuing system in steady state and therefore has no fixed set of users. Instead, users with certain parameters continuously arrive and depart. We assume that the users act rationally; thus,

the provider's strategies can be fully analyzed by backwards induction from the equilibria of the steady state of the queuing system.

Remark. *As we analyze the queuing system in steady state, our model works directly on the stochastic processes and not on individual realizations. Thus, all outcomes of interest, such as the provider's profit or the users' waiting times and payments (which we will introduce in the next sections), are always "in expectation," even if we do not always explicitly denote them accordingly.*

The remainder of this section is structured as follows: First we introduce the models for the provider (Section 3.1) and for the users (Section 3.2). Then we present the models for how the fixed-price and spot markets work in Sections 3.3 and 3.4, respectively.

3.1 Provider Model

The type of a provider is defined by a tuple $(\kappa_F, \kappa_L, T, l, \psi_E)$. Here, κ_F denotes the *fixed costs* an instance causes per time unit in the provider's cloud computing center, i.e., the total fixed costs the instance causes over its lifetime amortized per time unit. We can think of this as mainly hardware, infrastructure, and maintenance costs that are independent of the actual utilization. Conversely, κ_L denotes all *load-dependent costs* that an instance causes per time unit it is running. This overwhelmingly consists of increased electricity costs. We call the sum of fixed and load-depended costs the *instance costs* $\kappa := \kappa_F + \kappa_L$. T is an internal *SLA* (Service Level Agreement) for the fixed-price market that ensures a satisfactory quality of service. The SLA is said to be satisfied if the expected time until a newly arriving job in the fixed-price market starts running is below T.[8] In practice, the choice of T is influenced by many factors outside our model. We therefore assume T to be exogenously given and not to be part of the provider's strategy space, but our results hold for any T. The number of idle instances the provider has available for the spot market, and thus the maximum number of instances she could sell on the spot market, is denoted by l. We assume that the provider draws these idle instances from any part of the cloud computing center (e.g., maintenance instances, long-term reserved instances), except from those instances offered on the fixed-price market.[9] As these instances are already part of the provider's cloud computing center, they do not incur fixed costs κ_F a second time when offered on the spot market but they still incur load-dependent costs κ_L.

Since the steady state analysis is only concerned with mean service times, we do not specifically model a process by which idle capacity becomes available and unavailable. Instead, we let any $l' \leq l$ denote the l' idle instances with the lowest individual probabilities to become unavailable. $\psi_E(l')$ then denotes the expected number of times an instance randomly selected from among these l' instances becomes unavailable per time unit. Note that this makes $\psi_E(l')$ weakly increasing in l'. We set $\psi_E(0) = 0$ by convention. Obviously, whenever an instance that is currently running becomes unavailable, a user gets preempted. We call $\psi_E(l')$ the number of *external preemptions* because it only encompasses preemptions caused by the unreliable availability of idle instances. In addition to this, there are *internal preemptions* caused by changes in the current market price (which we introduce in Section 3.4).

[8]Note that the SLA can also take different shapes in practice and our model can easily be modified such that, instead of a limit on the expected queuing time T, some threshold on the percentage of rejections has to be met. Another possibility is to assume that users who cannot be served instantly simply do not join at all. Neither of these modifications change our main results.

[9]In Appendix D, we also provide an analysis for the case where the provider instead draws the idle instances from the *fixed-price market*. This introduces additional cross channel effects between both markets. Specifically, the number of available instances for the spot market as well as the external preemptions then directly depend on the users who choose the fixed-price market.

The choices of the provider define the setting in which the users find themselves. A strategy for the provider consists of a tuple $\rho = (p_F, l_S)$. We let p_F denote the price any user joining the fixed-price market has to pay per time unit his job is running. $l_S < l$ is the number of instances the provider decides to offer on the spot market.[10] $l_S = 0$ denotes that she decides to only offer a fixed-price market. For simplicity, we assume that the provider does not set a reserve price for the spot market. Introducing a reserve price would only strengthen our results as it expands the provider's strategy space to make the spot market more profitable.

Note that while the number of fixed-price instances l_F could technically be seen as part of the provider's strategy, the fixed-price market *must* satisfy the SLA T. This bounds the number of fixed-price instances from below. To keep our arguments simple, we assume that the provider will always offer the smallest number of fixed-price instances for a given user strategy profile σ (as defined in Section 3.4) such that the SLA is satisfied, as this minimizes her costs.[11] Consequently, in our model, l_F is not itself part of the provider strategy, but given as a function $l_F(\rho, \sigma)$ of the provider strategy ρ and user strategy profile σ.

Given provider strategy ρ and user strategy profile σ, we let $R_F(\rho, \sigma)$ denote the revenue from all fixed-price instances and $C_F(\rho, \sigma)$ denote the costs from all fixed-price instances. Similarly, $R_S(\rho, \sigma)$ and $C_S(\rho, \sigma)$ denote the revenue and cost from all spot instances. The provider's (expected) *profit per time unit* is then defined as the sum of her revenues minus her costs, i.e.,

$$\Pi(\rho, \sigma) := R_F(\rho, \sigma) + R_S(\rho, \sigma) - C_F(\rho, \sigma) - C_S(\rho, \sigma). \tag{1}$$

3.2 User Model

We model the resulting game for the users, given provider strategy ρ, as a queuing system. We assume this system to be in steady state (i.e., the state probabilities do not change over time). Thus, there is no fixed set of players, because users arrive and depart over time. This allows us to analyze strategies for every parameter set a user's *job* could have instead of having to artificially enumerate each individual user. Queuing theory provides us with tools to analyze (a) the time a newly arriving user has to wait until he gets to run his job and (b) his expected payment.

Formally, let there be *n job classes* with fixed *values* $v = (v_1, \ldots, v_n)$ for completion where $v_i > v_{i+1}$ for all $i \in \{1, \ldots, n-1\}$. New jobs from each class arrive sequentially according to a memoryless arrival process.[12] The *arrival rates* of the different job classes are $\lambda = (\lambda_1, \ldots, \lambda_n)$; i.e., in expectation, λ_i jobs of class i arrive per time unit. Each individual job requires exactly one instance to run and is associated with a distinct *user*. Users are only identified by the parameters of their jobs; the terms "user" and "job" can thus be used interchangeably. The *service time* for each job (i.e., the time it has to run on an instance assuming it is not preempted) is independently drawn according to a distribution with expectation $\frac{1}{\mu}$. To keep our expositions and proofs concise, we assume that all classes of jobs have the same mean service time. Our main results do not require specific service processes, as they only make use of the first moments of the distributions.[13] For every job of class i that arrives, a *waiting cost* c is independently and privately drawn from a distribution $F_i(c)$. This

[10] To keep the exposition simple and avoid special handling of corner cases, our formal model allows fractional instances (in both markets). As the cost of a single instance is negligible in realistic settings, this is a reasonable abstraction.

[11] Note that in practice, cloud providers can approximately follow such a cost minimization strategy because of the high turn-over rate of their hardware.

[12] This assumption is natural, especially for large cloud computing centers, and is supported by empirical studies (Zaharia et al. 2010, Zheng et al. 2016).

[13] The queuing theory literature often denotes this combination of memoryless arrival and general (independent) service processes as *M/GI/"number of instances"*.

distribution has a strictly positive PDF $f_i(c)$ on $[0, \mu v_i]$.[14] The waiting cost is incurred once per time unit until job completion. Every time a job is preempted, its user further incurs *preemption costs* in the form of an additional expected time loss τ. Concretely, this means that the expected payoff of a user with waiting cost c decreases by $c\tau$ for every expected preemption. Again, to keep the proofs concise, we assume that τ is independent of the job class. Note that when $\tau\psi_E(l_S) \geq 1$, then the time loss due to preemption per time unit is larger than one time unit (during which the job can again be preempted), such that (in expectation) the job will need an infinite amount of time to run to completion. Since that would trivially make offering a spot market of size l_S meaningless (as no user ever joins), we assume w.l.o.g. $\tau\psi_E(l) < 1$ (recall that l denotes the *maximum* number of spot instances the provider can offer). As is common in queuing theory, we assume that each job is infinitesimally small and does not affect the system dynamics on its own. We call the tuple of exogenous parameters and functions $(n, v, \lambda, \mu, F, \tau, \kappa_F, \kappa_L, T, l, \psi_E)$ a *setting*. The setting is assumed to be fixed and known by the provider and all users.

For any single user, a possible strategy consists of the tuple (α, β). $\alpha \in \{\mathcal{F}, \mathcal{S}, \mathcal{B}\}$ represents the decision whether to join the fixed-price market $\mathcal{F}$, the spot market $\mathcal{S}$ or to *balk* $\mathcal{B}$ (i.e., not to join any market and obtain zero payoff). To simplify notation, we assume that the action $\mathcal{S}$ is equivalent to balking when there is no spot market, i.e., when the provider sets $l_S = 0$. Further, any user submits a bid β for the spot market (which, if he joins the spot market, determines how quickly he gets an instance and how much he has to pay). For users who do not join the spot market, this bid has no effect, and thus, w.l.o.g., is set to be equal to their waiting cost c. The current state of the queues (i.e., which other users are currently in the system) is unobservable for users and thus cannot influence their strategies. A strategy profile σ encodes the strategies for any possible user. It consists of functions $\sigma_i : [0, \mu v_i] \to \{\mathcal{F}, \mathcal{S}, \mathcal{B}\} \times \mathbb{R}$, one for each class of jobs $i \in \{1, \ldots, n\}$, that map waiting costs c to strategies (α, β). Whenever a provider strategy ρ is given, a strategy profile with an asterisk (i.e., σ^*) denotes a corresponding equilibrium strategy profile for the users.

For any user, we now denote by $q(\alpha, \beta, \rho, \sigma)$ the expected *queuing time* (i.e., the time he spends in a queue without running his job on an instance) when he plays strategy (α, β), assuming provider strategy ρ and that all other users play according to the strategy profile σ. By $r(\alpha, \beta, \rho, \sigma)$ we denote the expected *running time* a user requires, i.e., the total time the user's job has to run on an instance until completion. $r(\alpha, \beta, \rho, \sigma)$ is the sum of the user's "normal" service time $\frac{1}{\mu}$ and the additional time his job requires because of preemptions. The expected *total waiting time* until job completion is the sum of queuing time and running time: $w(\alpha, \beta, \rho, \sigma) := q(\alpha, \beta, \rho, \sigma) + r(\alpha, \beta, \rho, \sigma)$. The user has to pay some amount of money for using an instance. We denote this expected *payment* $m(\alpha, \beta, \rho, \sigma)$. Overall, the expected *payoff* for a user of class i with waiting cost c is then given by $\pi_i^c(\alpha, \beta, \rho, \sigma) := v_i - cw(\alpha, \beta, \rho, \sigma) - m(\alpha, \beta, \rho, \sigma)$ for joining a market, and zero for balking.

Remark. *Our motivation for the notational separation of the 6 parameters of the payoff function $\pi_i^c(\alpha, \beta, \rho, \sigma)$ is as follows. The parameters i and c fully identify an individual user, where i denote this user's job class and c denotes this user's realized waiting cost. The remaining parameters $(\alpha, \beta, \rho, \sigma)$ all represent strategies.*

3.3 Fixed-price Market and Queue

The fixed-price market consists of a queuing system where users pay a fixed price p_F for every time unit their job is running. This results in an expected payment of $m(\mathcal{F}, \beta, \rho, \sigma) = \frac{p_F}{\mu}$. Since users do not get preempted in the fixed-price market, their running time is equal to their service time, i.e., $r(\alpha, \beta, \rho, \sigma) = \frac{1}{\mu}$. In contrast to Abhishek et al. (2017), we assume that sometimes, users

[14] Note that jobs with waiting costs $c > \mu v_i$ could only ever expect a negative payoff, even if they run instantly and pay nothing, and thus do not have to be considered.

have to wait until their job finds a free instance and begins running, leading to a short (expected) queuing time $T > 0$.[15] Recall that T is the SLA, which is given exogenously.

Remark. *An (expected) queuing time of $T = 0$ is not attainable with any finite number of instances. An infinite number of instances would not be realistic and makes any profit analysis meaningless, as the costs would also be infinite.*

The expected payoff of a user of class i with waiting cost c that joins the fixed-price market is thus equal to:

$$\pi_i^c(\mathcal{F}, \beta, \rho, \sigma) = v_i - cw(\mathcal{F}, \beta, \rho, \sigma) - m(\mathcal{F}, \beta, \rho, \sigma) \tag{2}$$

$$= v_i - c(T + \frac{1}{\mu}) - \frac{p_F}{\mu}. \tag{3}$$

Note that, in the fixed-price market, the user's payoff (i.e., Equation (3)) is independent of the actions of other users because it only depends on the provider's choice for the price p_F.

The provider's revenue $R_F(\rho, \sigma)$ in the fixed-price market is straightforwardly given by the arrival rate of users into the market, while the costs $C_F(\rho, \sigma)$ additionally depend on how many instances she has to offer in order to guarantee the SLA T:

$$R_F(\rho, \sigma) = \frac{p_F}{\mu} \left(\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{F}} f_i(x)dx \right) \tag{4}$$

$$C_F(\rho, \sigma) = \frac{\kappa_L}{\mu} \left(\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{F}} f_i(x)dx \right) + \kappa_F l_F(\rho, \sigma) \tag{5}$$

3.4 Spot Market and Queue

Following Abhishek et al. (2017), we model the spot market as a preemptible priority queue where both payments and the order in which jobs are run depend on the users' bids. The preemptible priority queue consists of l_S instances, running jobs in a priority-based order, where l_S is set as part of the provider's strategy. A job's priority is set by the bid given on arrival. A running job may be preempted for two different reasons. First, the job may be outbid by a job with a higher bid (*internal preemption*). Second, the instance the job is running on may become unavailable for some exogenous reason (like the instance being required for its primary purpose), independent of the job's priority (*external preemption*). The expected number of external preemptions per time unit $\psi_E(l_S)$ is independent of bids or other users and only depends on the provider's strategy. In contrast, how often a user's job gets *internally* preempted depends on the arrival rate of users with higher bids. We let $\psi_I(c, \rho, \sigma)$ denote the expected number of times a user with bid c in the spot market gets internally preempted, i.e., outbid by other users during a time unit. Note that ψ_I is fully determined by the queuing model and thus, in contrast to $\psi_E(l_S)$, it is not part of the setting. To summarize, a user's running time also depends on how often he gets preempted and how much time he loses with each preemption. The following proposition formally shows all of these dependencies.

Proposition 3.1. *A user's running time with bid c is*

$$r(\mathcal{S}, c, \rho, \sigma) = \begin{cases} \frac{1}{\mu} \frac{1}{1-\tau(\psi_I(c,\rho,\sigma)+\psi_E(l_S))} & \text{for } \tau(\psi_I(c, \rho, \sigma) + \psi_E(l_S)) < 1 \\ \infty & \text{otherwise} \end{cases} \tag{6}$$

[15] Note that it does not influence the equilibrium structure nor our results that jobs are queued and serviced in a first-come, first-served order. The same results are obtained if users instead continuously resubmit their jobs until they get a free instance (and are therefore effectively served in random order), because the expected service time would be the same.

Proof. During any time unit where his job is running, a user is on average preempted $(\psi_I(c,\rho,\sigma)+\psi_E(l_S))$ times, causing him to require an additional running time of $\tau(\psi_I(c,\rho,\sigma)+\psi_E(l_S))$. During this additional running time he is then on average again preempted $\tau(\psi_I(c,\rho,\sigma)+\psi_E(l_S))$ times. Summing over these recursive preemptions and multiplying by his service time $\frac{1}{\mu}$ yields a geometric series, i.e.,

$$r(\mathcal{S},c,\rho,\sigma) = \sum_{k=0}^{\infty} \frac{1}{\mu}\Big(\tau\Big(\psi_I(c,\rho,\sigma)+\psi_E(l_S)\Big)\Big)^k \tag{7}$$

$$= \begin{cases} \frac{1}{\mu}\frac{1}{1-\tau(\psi_I(c,\rho,\sigma)+\psi_E(l_S))} & \text{for } \tau(\psi_I(c,\rho,\sigma)+\psi_E(l_S)) < 1 \\ \infty & \text{otherwise} \end{cases} \tag{8}$$

□

Note that in any equilibrium, the running time of all jobs is trivially finite. Going forward, we can therefore safely ignore the case $\tau(\psi_I(c,\rho,\sigma)+\psi_E(l_S)) \geq 1$. A user's expected total waiting time when joining the spot market is therefore given by

$$w(\mathcal{S},c,\rho,\sigma) = q(\mathcal{S},x,\rho,\sigma) + \frac{1}{\mu}\frac{1}{1-\tau(\psi_I(c,\rho,\sigma)+\psi_E(l_S))}. \tag{9}$$

Payments in the spot market are set according to some spot market mechanism which we do not explicitly model, as we are only interested in the expected payment in equilibrium. Abhishek et al. (2017) showed that it suffices to analyze a Bayes Nash Incentive Compatible (BNIC) spot market mechanism for which a user's bid β only consists of a revelation of its true waiting cost c, i.e. $\beta = c$. In the following, we therefore use the terms bid, waiting cost and priority interchangeably.

From Lemma 5 of Abhishek et al. (2017), which is an adaptation of Myerson's famous Lemma (Myerson 1981) to spot markets, we know that for any BNIC market mechanism employed in the spot market, the expected payment has to be:

$$m(\mathcal{S},c,\rho,\sigma) = \int_0^c w(\mathcal{S},x,\rho,\sigma)dx - cw(\mathcal{S},c,\rho,\sigma) \tag{10}$$

We always assume that the provider employs a market mechanism whose payment rule for the spot market satisfies Equation (10). With such a payment rule, each user has to pay the difference between the overall cost caused by waiting he would incur at the mean waiting time of lower bids and the cost he incurs with his bid.

The total waiting time and the expected payment thus both depend on the number of users joining the spot market as determined by the strategy σ. For any user of class i with waiting cost c who joins the spot market the expected payoff can now be formulated as:

$$\pi_i^c(\mathcal{S},c,\rho,\sigma) = v_i - cw(\mathcal{S},c,\rho,\sigma) - m(\mathcal{S},c,\rho,\sigma) \tag{11}$$

$$= v_i - \int_0^c w(\mathcal{S},x,\rho,\sigma)dx. \tag{12}$$

The provider's revenue $R_S(\rho,\sigma)$ from the spot market now consists of the average payments users make. The costs $C_S(\rho,\sigma)$ are more complex, as a user getting preempted is also costly for the provider, since any job that loses time through preemption effectively has a longer running time and therefore causes more load-dependent costs. This means that the cost of the provider $C_S(\rho,\sigma)$

also depends on the number of preemptions:

$$R_S(\rho,\sigma) = \sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} m(\mathcal{S},x,\rho,\sigma) f_i(x)dx \tag{13}$$

$$C_S(\rho,\sigma) = \kappa_L \sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} \frac{1}{\mu}\frac{1}{1-\tau(\psi_I(x,\rho,\sigma)+\psi_E(l_S))} f_i(x)dx \tag{14}$$

4 EQUILIBRIUM ANALYSIS

In this section, we analyze the resulting Bayes Nash equilibria (BNEs) of the user game given a provider strategy ρ. A provider strategy can result in three basic types of equilibria: (a) *pure fixed-price equilibria*, where no user joins the spot market, (b) *pure spot equilibria*, where no user joins the fixed-price market and (c) *real hybrid equilibria*, where each market is chosen by some users. In the following, we first individually characterize the three types of equilibria (Sections 4.1, 4.2 and 4.3). In Section 4.4, we then show how to identify the type of equilibrium in which any given provider strategy ρ would result.

4.1 Pure Fixed-price Equilibria

As we have shown in Section 3.3, the payoff of a user in the fixed-price market does not depend on the actions of the other users. When no spot market is offered, the strategy of any user is therefore independent of the other users. The payoff $\pi_i^c(\mathcal{F},\beta,\rho,\sigma) = v_i - c(T+\frac{1}{\mu}) - \frac{p_F}{\mu}$ for submitting a job is a monotonically decreasing function of the waiting cost c for every job class. We can therefore easily show that all equilibrium strategy profiles take the form of threshold functions.

Proposition 4.1. *For any provider strategy $\rho = (p_F, 0)$, the users' equilibrium strategy profile in any BNE takes the form $\sigma^* = \overrightarrow{c}^F$. Here, overloading our previous notation, $\sigma = \overrightarrow{c}^F$ denotes that all users of class i with waiting cost $c < c_i^F$ join the fixed-price market, i.e., they play $\alpha = \mathcal{F}$, and all users of class i with waiting cost $c > c_i^F$ balk and obtain zero payoff. The cutoff vector $\overrightarrow{c}^F = (c_1^F, \ldots, c_n^F)$ is the unique solution to the following system of equations:*

$$c_i^F = \frac{v_i - p_F\frac{1}{\mu}}{(T+\frac{1}{\mu})} \quad \forall i \in \{1,\ldots,n\} \tag{15}$$

Proof. Whenever $l_S = 0$, it directly follows that any equilibrium strategy profile is defined by a cutoff vector $\sigma^* = \overrightarrow{c}^F$ by the monotonicity of the payoff $\pi_i^c(\mathcal{F},\beta,\rho,\sigma)$ in c. The expression for c_i^F follows via simple algebra by setting $\pi_i^{c_i^F}(\mathcal{F},\beta,\rho,\sigma) = v_i - c_i^F(T+\frac{1}{\mu}) - \frac{p_F}{\mu} = 0$. □

Remark. *With a strategy profile of the form $\sigma = \overrightarrow{c}^F$, users whose jobs have waiting costs $c = c_i^F$ can join the fixed-price market or balk and obtain zero payoff either way. We do not need to use a tie-breaking rule because these users constitute a set with measure zero and do not influence the provider's profit or any user's payoff. The same holds for all future strategy profile characterizations we present using cutoff vectors.*

4.2 Pure Spot Equilibria

Pure spot equilibria arise when at least some spot instances are offered and no user joins the fixed-price market. If the provider chooses the price per time unit for fixed-price instances too high, every user either joins the spot market or cannot obtain a positive payoff in either market and balks. For settings without preemption costs, these equilibria have previously been studied in Abhishek et al. (2017). We now provide an analogous result for settings with preemption costs.

Proposition 4.2. *For any provider strategy $\rho = (p_F, l_S)$ with $l_S > 0$, in any BNE of the user game where no user joins the fixed-price market, the equilibrium strategy profile has the form $\sigma^* = \overrightarrow{c}^S$. Here, $\sigma^* = \overrightarrow{c}^S$ denotes that all users of class i with waiting cost $c < c_i^S$ join the spot market, i.e., they play $\alpha = \mathcal{S}$ and truthfully bid c, and all users of class i with waiting cost $c > c_i^S$ balk and obtain zero payoff. The cutoff vector $\overrightarrow{c}^S = (c_1^S, \ldots, c_n^S)$ is the unique solution to the following system of equations:*

$$v_i - \int_0^{c_i^S} w(\mathcal{S}, x, \rho, \overrightarrow{c}^S)dx = 0 \quad \forall i \in \{1, \ldots, n\} \tag{16}$$

The proof of Proposition 4.2 can be found in Appendix A.2. Intuitively, the existence of a cutoff vector $\overrightarrow{c}^S$ follows because, for any fixed strategy profile σ, the payoff of a user of class i, i.e., $\pi_i^c(\mathcal{S}, c, \rho, \sigma)$, is monotone decreasing in his waiting cost c.

4.3 Hybrid Equilibria

We now analyze *hybrid equilibria* that arise when the provider plays $l_S > 0$ on the spot market and some users still join the provider's fixed-price market. These equilibria can again be subdivided into two cases. In the first case, jobs in the spot market incur relatively high preemption costs and (independent of their bid) take longer until completion than in the fixed-price market, i.e., $w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*) \leq w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*)$, where $\overline{\beta}$ denotes a bid that is higher than any other bid in σ^*. Plugging in the definitions and using simple algebra, this holds whenever $T \leq \frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right)$. To understand the second case, note that the jobs with the highest bids are started instantly in the spot market, while they always have to wait some small time $T > 0$ in expectation to start running in the fixed-price market. When spot instances are very reliably idle, this can in theory lead to situations where the spot market requires a shorter time until completion than the fixed-price market for jobs with very high bids (i.e. $w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*) > w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*)$). While in practice, this is unlikely to happen, we still need to include this case in the equilibrium analysis.

In the following, we first analyze the case where the spot market is slower than the fixed-price market. We then present the equilibrium analysis for the more exotic case where the spot market is faster for a small number of user.

4.3.1 *Case 1: Spot Market Slower Than the Fixed-price Market.*

If the overall time lost through external preemptions with the highest bid (i.e., without ever getting internally preempted) is higher than the expected queuing time in the fixed-price market, then no user is willing to pay more in the spot market than in the fixed-price market. As the overall costs (i.e., the costs for waiting plus the payment) of a user who has arrived into the system do not depend on his class, there exists a waiting cost for which both markets result in the same payoff, independent of a job's class. Below that waiting cost users prefer the spot market and above it they prefer the fixed-price market; though in either case they might still balk if their value is too low (leading to a negative payoff in both markets). This again allows us to state the equilibrium strategy profiles as cutoff vectors, this time with two vectors $\overrightarrow{c}^P$ (P for payoff equivalence) and $\overrightarrow{c}^B$ (B for balking).

Proposition 4.3. *For any provider strategy $\rho = (p_F, l_S)$ with $l_S > 0$ and $w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*) \leq w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*)$, in any BNE of the user game where any user joins the fixed-price market, the equilibrium strategy profile is of the form $\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$. Here, $\sigma = (\overrightarrow{c}^P, \overrightarrow{c}^B)$ denotes that a user of class i with waiting cost c joins the spot market when $c < c_i^P \leq c_i^B$ and the fixed-price market when $c_i^P < c < c_i^B$; when $c > c_i^B$, he balks and does not join any market. The cutoff point c_1^P and the cutoff vector $\overrightarrow{c}^B$ are*

the unique solution to the following system of equations:

$$0 = c_1^P(T + \frac{1}{\mu}) + \frac{p_F}{\mu} - \int_0^{c_1^P} w(\mathcal{S}, x, \rho, (\overrightarrow{c}^P, \overrightarrow{c}^B))dx \tag{17}$$

$$0 = v_i - \min\left\{c_i^B(\frac{1}{\mu} + T) + \frac{p_F}{\mu}, \int_0^{c_i^B} w(\mathcal{S}, x, \rho, (\overrightarrow{c}^P, \overrightarrow{c}^B))dx\right\} \quad \forall i \in \{1, \ldots, n\} \tag{18}$$

The rest of the cutoff vector $\overrightarrow{c}^P$ is given as $c_i^P = min(c_1^P, c_i^B)$.

The proof of Proposition 4.3 can be found in Appendix A.3. In words, $\overrightarrow{c}^P$ denotes until which waiting cost the payoff in the spot market is higher than either balking or joining the fixed-price market. The second vector (i.e., $\overrightarrow{c}^B$) denotes above which waiting cost neither market allows users to obtain a positive payoff anymore. Note that there are often some classes i for which $c_i^P = c_i^B$, i.e., no user from those classes joins the fixed-price market.

4.3.2 *Case 2: Spot Market Faster Than the Fixed-price Market.*

When the spot market for very high bids is faster than the fixed-price market, i.e., $w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*) < w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*)$, then some users are willing to pay more in the spot market than what they would in the fixed-price market. However, most users in the spot market still have to wait longer, and are therefore not willing to pay as much as in the fixed-price market. As a result, we obtain the following equilibrium strategy profiles, now with three cutoff vectors: $\overrightarrow{c}^L$ (L for lower bound of the fixed-price market), $\overrightarrow{c}^U$ (U for upper bound of the fixed-price market) and $\overrightarrow{c}^H$ (H for hybrid).

Proposition 4.4. *For any provider strategy $\rho = (p_F, l_S)$ with $l_S > 0$, in any BNE of the user game where any user joins the fixed-price market and where it holds that $w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*) < w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*)$, the equilibrium strategy profile is of the form $\sigma^* = (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$. Here, $\sigma = (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$ denotes that a user of class i with waiting cost c joins the spot or the fixed-price market if and only if $c \leq c_i^H$; out of those users that join any market, almost all (i.e., all besides possibly a set of measure zero that does not influence system dynamics) choose the fixed-price market if and only if $c_i^L \leq c \leq c_i^U$ and choose the spot market otherwise. The cutoff points c_1^L, c_1^U and the cutoff vector $\overrightarrow{c}^H$ are the unique solution to the following system of equations:*

$$c_1^L(T + \frac{1}{\mu}) + p_F\frac{1}{\mu} - \int_0^{c_1^L} w(\mathcal{S}, x, \rho, (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H))dx = 0 \tag{19}$$

$$c_1^U(T + \frac{1}{\mu}) + p_F\frac{1}{\mu} - \int_0^{c_1^U} w(\mathcal{S}, x, \rho, (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H))dx = 0 \tag{20}$$

$$v_i - \int_0^{c_i^H} w(\mathcal{S}, x, \rho, (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H))dx = 0 \quad \forall i \in \{1, \ldots, n\} \tag{21}$$

The rest of the cutoff vectors $\overrightarrow{c}^L$ and $\overrightarrow{c}^U$ are given as $c_i^L = min(c_1^L, c_i^H)$ and $c_i^U = min(c_1^U, c_i^H)$.

The proof of Proposition 4.4 can be found in Appendix A.5. In words, $\overrightarrow{c}^H$ denotes the waiting costs of each class above which users cannot obtain a positive payoff in either market and balk. Of those users that do not balk, only users of class i whose waiting cost c lies between c_i^L and c_i^U join the fixed-price market, while every other user whose waiting cost c lies below c_i^H joins the spot-market.

4.4 Equilibria Resulting From Provider Strategy

So far, we have analyzed different forms of equilibria for the user game. We now analyze what kind of equilibrium a given provider strategy ρ results in. Because users with very low waiting costs

always join the spot market, pure fixed-price equilibria can only result when no spot instances are offered, i.e., when the provider plays $l_S = 0$. Going forward, we call such provider strategies *fixed-price strategies*. Conversely, we call all strategies where the provider offers any spot instances (i.e., where she plays $l_S > 0$) *hybrid strategies*.

Given a hybrid strategy ρ, it is easy to differentiate which of the two different hybrid equilibria the strategy ρ potentially results in by simply checking whether a congestion free spot market is faster than the fixed-price market, i.e., by comparing T to $\frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right)$. But it is not directly apparent whether a hybrid strategy results in a "real" hybrid equilibrium or degenerates the market into a pure spot equilibrium where no user joins the fixed-price market. Fortunately, we can formulate a simple condition that, as we will show, distinguishes these two types of equilibria.

Definition 4.5 (Proper Hybrid Strategy). For any ρ with $l_S > 0$, let $\sigma = \overrightarrow{c}^S$ be a strategy profile satisfying Equation (16). Recall that, with such a strategy profile σ, no user joins the fixed-price market and σ would be an equilibrium strategy profile (of the spot market) if no fixed-price market existed. We say that ρ is a *proper hybrid strategy* (or *proper*) if one of the following holds:

(1) $w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*) \leq w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*)$ and the payoff for users with waiting cost c_1^S is higher in the fixed-price than in the spot market (and thus a beneficial deviation from $\sigma = \overrightarrow{c}^S$ exists), or
(2) $w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*) < w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*)$ and, under strategy profile $\sigma = \overrightarrow{c}^S$, there exists a waiting cost c', such that (a) the total waiting time with bid c' in the spot market is equal to the waiting time in the fixed-price market, and (b) the payoff for a user with waiting cost c' is higher in the fixed-price market than in the spot market (and thus a beneficial deviation from $\sigma = \overrightarrow{c}^S$ exists).

Informally, a *proper hybrid strategy* only requires that users with one specific waiting cost (i.e., c') prefer the fixed-price market over the spot market. The definition is well defined because Eq. (16) always has a solution and thus allows us to calculate the strategy profile $\overrightarrow{c}^S$. We now show that this definition enables us to determine whether or not a given provider strategy results in a BNE where some users join the fixed-price market.

Lemma 4.6. *Let $\rho = (p_F, l_S)$ be a hybrid strategy, i.e., a provider strategy with $l_S > 0$. In any BNE σ^* of the user game, some users join the fixed-price market (i.e., $\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$ or $\sigma^* = (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$) if and only if ρ is proper.*

The proof of Lemma 4.6 can be found in Appendix A.6.

With this result, we can now give a full equilibrium characterization result based purely on the provider strategy ρ.

Theorem 4.7. *For any provider strategy ρ, the equilibrium strategy profile of the users is*

(1) $\sigma^* = \overrightarrow{c}^F$ *if and only if ρ is a fixed-price strategy, i.e., $l_S = 0$.*
(2) $\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$ *if and only if ρ is a proper hybrid strategy and it holds that $w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*) \leq w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*)$.*
(3) $\sigma^* = (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$ *if and only if ρ is a proper hybrid strategy and it holds that $w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*) < w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*)$.*
(4) $\sigma^* = \overrightarrow{c}^S$ *otherwise.*

Proof. We show the "if" direction of each case in turn. Once the "if" direction for all cases have been shown, the "only if" direction immediately follows for each of them.

(1) Assume that $l_S = 0$, then $\sigma^* = \overrightarrow{c}^F$ follows from Proposition 4.1. For $l_S > 0$ at least users with waiting costs in some small neighborhood around zero will trivially prefer the spot market.

(2) Assume that $l_S > 0$, $w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*) \leq w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*)$ and the strategy is proper. By Lemma 4.6 some users join the fixed-price market. By Proposition 4.3 it follows that $\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$.
(3) Assume that $l_S > 0$, $w(\mathcal{S}, \overline{\beta}, \rho, \sigma^*) < w(\mathcal{F}, \overline{\beta}, \rho, \sigma^*)$ and the strategy is proper. By Lemma 4.6 some users join the fixed-price market. By Proposition 4.4 it follows that $\sigma^* = (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$.
(4) Assume that $l_S > 0$ and the strategy is not proper. By Lemma 4.6 no users join the fixed-price market. By Proposition 4.2 it follows that $\sigma^* = \overrightarrow{c}^S$.

□

5 PUTTING IT ALL TOGETHER: PROVIDER PROFIT AND USER WELFARE

In the previous section, we have derived the user equilibrium strategy profiles σ^*, given different provider strategies ρ. As these uniquely define the provider's costs and revenue, we can now bound how a provider's costs change when she offers a spot market. Using these bounds, we then assemble a condition that only depends on the setting and show that, when this condition holds, a provider can always simultaneously increase her profit and achieve a Pareto improvement for the users by also offering a spot market alongside her fixed-price market.

5.1 Bounding the Provider's Costs

Towards deriving our results, we now bound the costs of the spot and fixed-price markets. First, we state a condition to formally bound the fixed-cost savings a provider can obtain by offering spot instances. Second, we derive a lemma to bound the average running time in the spot market. We then use these to bound the provider's costs. We begin with bounding the reduction in fixed-price instances. Going forward, we denote as σ_0^* the user equilibrium resulting from a provider strategy with $l_S = 0$ spot instances to distinguish it from the user equilibrium σ^* resulting from another strategy.

Lemma 5.1. *For any fixed-price strategy $\rho_0 = (p_F^0, 0)$ and any hybrid strategy $\rho = (p_F^0, l_S)$ with the same price p_F^0, denote by $\Delta l_F(l_S) := l_F(\rho_0, \sigma_0^*) - l_F(\rho, \sigma^*)$ the reduction in required fixed-price instances and by $\lambda_S(l_S)$ the arrival rate of all users who join the fixed-price market under ρ_0 but move to the spot market if the provider offers l_S spot instances. Then it holds that*

$$\Delta l_F(l_S) \geq \frac{1}{\mu}\lambda_S(l_S). \tag{22}$$

Proof. Given that the arrival process into the system (i.e., before users take an action) is memoryless (i.e., the number of arrivals in any time interval is distributed as a Poisson random variable), the arrival process into the fixed-price market for any strategy profile σ is also memoryless. Furthermore, the number of users who choose the fixed-price market in equilibrium is strictly decreasing in the number of offered spot instances l_S. Thus, in equilibrium, the fixed-price markets under any ρ_0 and ρ can be seen as two queues Q_1, Q_2 with the same queuing time, Poisson arrival rates $\lambda_1 > \lambda_2$, the same service process, and l_1, l_2 instances, respectively. To keep the notation simple, w.l.o.g., we assume that the service time is normalized to $\frac{1}{\mu} = 1$. The statement of the lemma is therefore equivalent to the difference between the number of instances of any such Q_1, Q_2 being larger or equal to the difference between the arrival rates, i.e., $l_1 - l_2 \geq \lambda_1 - \lambda_2$.

Since $\lambda_1 > \lambda_2$, we can write $\lambda_1 = \lambda_A + \lambda_2$ for some $\lambda_A > 0$. Since the arrival process into Q_1 is memoryless, we can further see it as a mix of two independent arrival processes A and B with rates λ_A and $\lambda_B = \lambda_2$. Now, for the sake of contradiction, assume that $l_1 - l_2 < \lambda_1 - \lambda_2 = \lambda_A$. Since the number of instances l_1 is finite, some users will sometimes have to wait to be served. Thus, at any randomly chosen point in time, there are, in expectation, strictly more than λ_A users that arrived from arrival process A in Q_1. Given that $l_1 - \lambda_A < l_2$, there are, at any random point in

time (in expectation) less than l_2 instances available to serve the users from arrival process B in Q_1. Given that the arrival processes are memoryless, the distribution of the states of the queue at a random point in time is the same as whenever a random user arrives.[16] Therefore, there are also, in expectation, less than l_2 instances available for users from process B whenever a random user from process B arrives. This implies that when a random user arrives from process B into Q_1, in expectation, there are strictly fewer idle instances available in Q_1 than when a random user arrives into Q_2. It follows that users from arrival process B (in expectation) have a longer queuing time in Q_1 than users have in Q_2. Since Q_1 is a FIFO queue with memoryless arrivals, every user in Q_1 has the same expected queuing time independent of whether he arrives by process A or B. Consequently, the queuing time of any user in Q_1 is longer than the queuing time of any user in Q_2, a contradiction to our definition of Q_1 and Q_2. Thus, it must hold that $l_1 - l_2 \geq \lambda_1 - \lambda_2$. □

Note that Equation (22) from Lemma 5.1 immediately implies a lower bound on the fixed costs the provider saves by offering a spot market. This is the case because the provider's fixed costs only depend on κ_F and the number of fixed-price instances required in equilibrium.

Next, we bound the average running time in the spot market.

Lemma 5.2. *The average running time in the spot market (i.e., the left-hand side of the following inequality) is bounded above as follows:*

$$\frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} r(\mathcal{S},x,\rho,\sigma) f_i(x)dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x)dx} < (\frac{1}{\mu}+\tau)\frac{1}{1-\tau\psi_E(l_S)} \tag{23}$$

Proof. First, note that in any queue that is stable, i.e., where every user has a finite waiting time, (on average) the exact same number of users arrive and depart per time unit. Now assume that any arriving user preempts one other job through its arrival. Obviously, this is an upper bound, as no job can preempt more than one job due to its arrival. Since the average number of arrivals is the same as the average number of departures, after a job arrives and causes a preemption, another job has to depart (in expectation) before the next preemption. Thus, (on average) a job cannot be internally preempted more than once during its whole running time. Denoting the number of internal preemptions a job suffers in expectation by $\psi_I(c,\rho,\sigma)r(\mathcal{S},c,\rho,\sigma)$, it follows that:

$$\frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} \psi_I(x,\rho,\sigma) r(\mathcal{S},x,\rho,\sigma) f_i(x)dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x)dx} < 1 \tag{24}$$

Now note that the full running time of a job can be split into the running time it would require without internal preemption and any additional running time $r_I(\mathcal{S},c,\rho,\sigma)$ needed because of internal preemption (caused either directly, or indirectly via additional external preemptions that occur during the additional running time), i.e.,

$$r(\mathcal{S},c,\rho,\sigma) = (\frac{1}{\mu}\frac{1}{1-\tau\psi_E(l_S)}) + r_I(\mathcal{S},c,\rho,\sigma). \tag{25}$$

While the time lost directly through internal preemptions is given by $\psi_I(c,\rho,\sigma)r(\mathcal{S},c,\rho,\sigma)\tau$, additional external preemptions can occur during this time. By the same argument as for the

[16] This is called the PASTA (Poisson Arrivals See Time Averages) property (see (Wolff 1982)).

running time in Proposition 3.1 (i.e., viewing it as a geometric series) it follows that

$$r_I(\mathcal{S}, c, \rho, \sigma) = \sum_{k=0}^{\infty} \psi_I(c, \rho, \sigma) r(\mathcal{S}, c, \rho, \sigma) \tau (\tau \psi_E(l_S))^k \tag{26}$$

$$= \psi_I(c, \rho, \sigma) r(\mathcal{S}, c, \rho, \sigma) \tau \frac{1}{1 - \tau \psi_E(l_S)}. \tag{27}$$

We can now upper bound the average additional running time caused by internal preemptions:

$$\frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} r_I(\mathcal{S}, x, \rho, \sigma) f_i(x) dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x) dx} \tag{28}$$

$$= \frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} \psi_I(x, \rho, \sigma) r(\mathcal{S}, x, \rho, \sigma) \tau \frac{1}{1-\tau\psi_E(l_S)} f_i(x) dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x) dx} \tag{29}$$

$$= \frac{\tau}{1 - \tau \psi_E(l_S)} \frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} \psi_I(x, \rho, \sigma) r(\mathcal{S}, x, \rho, \sigma) f_i(x) dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x) dx} \tag{30}$$

$$< \frac{\tau}{1 - \tau \psi_E(l_S)}. \tag{31}$$

Here, the inequality in (31) follows by plugging (30) into Inequality (24). It now directly follows for the average running time that

$$\frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} r(\mathcal{S}, x, \rho, \sigma) f_i(x) dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x) dx} \tag{32}$$

$$= \frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} (\frac{1}{\mu} \frac{1}{1-\tau\psi_E(l_S)}) f_i(x) dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x) dx} + \frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} r_I(\mathcal{S}, x, \rho, \sigma) f_i(x) dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x) dx} \tag{33}$$

$$< \frac{1}{\mu} \frac{1}{1 - \tau \psi_E(l_S)} + \frac{\tau}{1 - \tau \psi_E(l_S)} \tag{34}$$

$$= (\frac{1}{\mu} + \tau) \frac{1}{1 - \tau \psi_E(l_S)}. \tag{35}$$

□

Lemma 5.2 immediately implies a bound on the load-dependent costs incurred from offering a spot market by combining Equation (23) with the definition of the costs $C_S(\rho, \sigma)$ in Equation (14).

5.2 Well-behaved Settings: Increasing Provider Profit and User Welfare

In this subsection, we use Lemmas 5.1 and 5.2 to derive a very mild condition on the setting under which we then show our main result. First, recall that the average running time in the fixed-price market is equal to the service time $\frac{1}{\mu}$. As we have shown in Lemma 5.2, the average running time in the spot market is bounded by $(\frac{1}{\mu} + \tau) \frac{1}{1-\tau\psi_E(l_S)}$. This immediately implies an upper bound on the *difference* between the average running time in the spot and the fixed-price market: $\frac{1}{\mu} \left(\frac{1+\tau\mu}{1-\tau\psi_E(l_S)} - 1 \right)$. The following condition puts this bound (after normalizing by the service time $\frac{1}{\mu}$) in relation to the ratio between the provider's fixed and load-dependent costs.

Definition 5.3. We call a setting $(n, v, \lambda, \mu, F, \tau, \kappa_F, \kappa_L, T, l, \psi_E)$ *well-behaved* if there exists a number of spot instances l^w with $0 < l^w \leq l$ such that the following holds:

$$\frac{1+\tau\mu}{1-\tau\psi_E(l^w)} - 1 < \frac{\kappa_F}{\kappa_L} \tag{36}$$

In the following theorem, we show that, in a well-behaved setting, a provider can increase her profit as well as achieve a Pareto improvement for the users by offering a spot market, compared to only offering a fixed-price market.

Theorem 5.4. *Given a well-behaved setting, for every fixed-price strategy $\rho_0 = (p_F^0, 0)$ that results in a positive profit, there exists a hybrid strategy $\rho = (p_F^0, l_S)$ with the same price p_F^0 and with $0 < l_S \leq l$ that yields a higher profit for the provider, i.e.,*

$$\Pi((p_F^0, l_S), \sigma^*) > \Pi((p_F^0, 0), \sigma_0^*), \tag{37}$$

and the same strategy also yields a Pareto improvement for the users, i.e.,

$$\forall i \in \{1, \dots, n\}\ \forall c \in [0, \mu v_i] : \quad \pi_i^c(\alpha, \beta, \rho, \sigma^*) \geq \pi_i^c(\alpha, \beta, \rho_0, \sigma_0^*), \text{ and} \tag{38}$$

$$\exists i \in \{1, \dots, n\}\ \exists c \in [0, \mu v_i] : \quad \pi_i^c(\alpha, \beta, \rho, \sigma^*) > \pi_i^c(\alpha, \beta, \rho_0, \sigma_0^*). \tag{39}$$

Proof. First, we collect all required auxiliary results. Denote by $\lambda_S(l_S)$ the arrival rate of all users who join the fixed-price market under σ_0^* but move to the spot market under σ^*. Denote by $\lambda_N(l_S)$ the arrival rate of all users who balk under σ_0^* but newly join the spot market under σ^*. Thus, the arrival rate of all users into the spot market under σ^* is $\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x)dx = \lambda_N(l_S) + \lambda_S(l_S)$.

Recall that Lemma 5.1 directly translates to a lower bound on the difference in the fixed costs incurred under ρ and ρ_0, i.e., $\kappa_F \Delta l_F(l_S) \geq \kappa_F \frac{1}{\mu}\lambda_S(l_S)$. Denote by $b(l_S)$ the bound on the normalized average running time as derived in Lemma 5.2, i.e., $b(l_S) := \frac{1+\tau\mu}{1-\tau\psi_E(l_S)}$. By combining Equation (23) with the definition of $C_S(\rho, \sigma)$ in Equation (14), it then follows that the increase in load-dependent costs caused by $\lambda_S(l_S)$ users switching from the fixed-price to the spot market and $\lambda_N(l_S)$ users switching from balking to the spot market is bounded by $\frac{\kappa_L}{\mu}(\lambda_S(l_S) - (\lambda_N(l_S) + \lambda_S(l_S))b(l_S))$. We can now use both of these results to bound the cost difference between ρ_0 and ρ:

$$C_F(\rho_0, \sigma_0^*) - \big(C_F(\rho, \sigma^*) + C_S(\rho, \sigma^*)\big) \geq \frac{\kappa_F}{\mu}\lambda_S(l_S) + \frac{\kappa_L}{\mu}(\lambda_S(l_S) - (\lambda_N(l_S) + \lambda_S(l_S))b(l_S)) \tag{40}$$

$$= \lambda_S(l_S)\left(\frac{\kappa_F}{\mu} + \frac{\kappa_L}{\mu}(1 - b(l_S))\right) - \lambda_N(l_S)\frac{\kappa_L}{\mu}b(l_S) \tag{41}$$

Lemma A.3 in the appendix shows that for every $\varepsilon > 0$, there exists an l_S such that the *average* payment of the users that join the spot market $\bar{m}(\mathcal{S}, (p_F^0, l_S), \sigma^*)$ is at least the expected payment in the fixed-price market minus ε, i.e.,

$$\bar{m}(\mathcal{S}, (p_F^0, l_S), \sigma^*) \geq \frac{p_F^0}{\mu} - \varepsilon. \tag{42}$$

For any $l_S \leq l^w$ that satisfies Equation (42) for a given ε, the revenue difference between ρ and ρ_0 can therefore be bounded as follows:

$$R_S((p_F^0, l_S), \sigma^*) + R_F((p_F^0, l_S), \sigma^*) - R_F((p_F^0, 0), \sigma_0^*) \tag{43}$$

$$= (\lambda_S(l_S) + \lambda_N(l_S))\left[\bar{m}(\mathcal{S}, (p_F^0, l_s), \sigma^*)\right] - \lambda_S(l_S)\left\lceil \frac{p_F^0}{\mu} \right\rceil \tag{44}$$

$$\geq -\lambda_S(l_S)\varepsilon + \lambda_N(l_S))\left\lceil \frac{p_F^0}{\mu} - \varepsilon \right\rceil \tag{45}$$

Combining the bounds on revenue and costs for any $l_S \leq l^w$ that satisfies Equation (42) for a given ε, the profit difference between the hybrid strategy $\rho = (p_F^0, l_S)$ and the fixed-price strategy $\rho_0 = (p_F^0, 0)$ can now be bounded as follows:

$$\Pi((p_F^0, l_S), \sigma^*) - \Pi((p_F^0, 0), \sigma_0^*) \tag{46}$$

$$\geq \lambda_S(l_S) \left[\frac{\kappa_F}{\mu} + \frac{\kappa_L}{\mu}(1 - b(l_S)) - \varepsilon \right] + \lambda_N(l_S) \left[\frac{p_F^0}{\mu} - \varepsilon - \frac{\kappa_L}{\mu} b(l_S) \right] \tag{47}$$

To see that this expression is positive for correctly chosen ε and l_S, note that for well-behaved settings, it directly follows that

$$\frac{\kappa_F}{\mu} + \frac{\kappa_L}{\mu}(1 - b(l_S)) > 0. \tag{48}$$

Since offering more spot instances causes a higher external preemption rate, the left-hand side of Equation (48) decreases in l_S. Further, since the fixed-price strategy obtains a positive profit, it also holds that

$$\frac{p_F^0}{\mu} > \frac{\kappa_F}{\mu} > \frac{\kappa_L}{\mu} b(l_S). \tag{49}$$

Thus, for ε small enough, any strategy $\rho = (p_F^0, l_S)$ with l_S satisfying Equation (42) increases the profit compared to the fixed-price strategy $\rho_0 = (p_F^0, 0)$. Since the price p_F^0 did not change, the users still have access to the same fixed-price market as before, but additionally now also have access to a spot market (which some users prefer), which leads to a Pareto improvement for the users. □

Theorem 5.4 shows that in any well-behaved setting, a provider can increase her profit and at the same time achieve a Pareto improvement for the users by playing a hybrid strategy compared to playing a fixed-price strategy. The following corollary follows immediately.

Corollary 5.5. *In any well-behaved setting, the provider's profit-optimal strategy is a hybrid strategy.*

Remark. *Note that the strategies characterized by Theorem 5.4 and Corollary 5.5 are typically not the same. In particular, while the strategy described by Theorem 5.4 simultaneously increases the provider's profit and leads to a Pareto improvement for the users, the profit-optimal strategy described by Corollary 5.5 may increase or* decrease *user welfare.*

5.3 Discussion

Our results show that, in a well-behaved setting, a cloud provider can increase her profit by offering a spot market consisting of idle capacity in addition to offering her existing fixed-price market. Informally, *well-behavedness* guarantees that the additional load-dependent costs incurred due to running a spot market are lower than the fixed costs saved through utilizing idle capacity. For a setting to be well-behaved, the idle capacity used for the spot market has to be sufficiently reliably idle and individual preemptions have to be sufficiently inexpensive compared to the ratio between fixed and load-dependent costs. In practice, a cloud provider can easily check this, as the required condition is independent of internal preemptions and only depends on setting parameters. The fixed and load-dependent costs κ_F and κ_L, the preemption costs τ, the expected service time $\frac{1}{\mu}$, and the rate of external preemptions ψ_E (i.e., how reliably idle the provider's capacity is) do not depend on the dynamics of the queue. Whether a setting is well-behaved can therefore directly be evaluated without having to use any queuing-theoretic formulas or equilibrium calculations.

Our well-behavedness condition is quite mild and most cloud providers should find it satisfied in practice. To see this, note that fixed costs κ_F are usually about 5 to 20 times higher than load-dependent costs (Barroso et al. 2018). Even if each preemption in the spot market resulted in an additional run time of 25% of a job's expected service time (which is an unreasonably high number, considering that mostly users with low preemption costs τ would use the spot market), the condition would still be satisfied if a job on average gets externally preempted 3 times per time unit before it finishes. Of course, some cloud providers may see the condition *not* satisfied: for example, if they have very low fixed costs (e.g., by using old instances whose acquisition costs are already amortized) or if they do not have reliably idle capacity. This demonstrates how our condition can inform the provider's managerial decision making process.

A provider whose setting is well-behaved might also want to compute her optimal strategy, i.e., the optimal price and number of spot instances. Unfortunately, directly calculating the optimal strategy is only feasible for very few arrival/service processes (see Section 6 for examples). For general service processes, formulas for queuing times are open problems of queueing theory. However, this does not mean that a provider cannot find a profit-increasing hybrid strategy. In practice, the provider can keep the same price she used when only offering fixed-price instances and start with a relatively small spot market. According to Theorem 5.4, this already leads to a (small) profit increase. Over time, the provider can then successively increase the size of the spot market until she observes no further profit increase. Alternatively, she can employ a reserve price for the spot market, starting with a relatively high price and successively decreasing it.

Note that in this work, we have analyzed the profit per time unit, and therefore our model does not include *one-time costs*. However, depending on how their cloud computing centers are structured, providers might face varying degrees of one-time costs (e.g., to set up a new marketplace and enable offering preemptible instances). While these costs are only incurred once and therefore become negligible over time, a provider with a shorter planning horizon might still want to take these costs into account. Providers may also face additional costs whenever a user gets preempted (e.g., for re-booting a machine after a preemption). As these costs can take different shapes for different providers and do not influence the user sub-game, we did not include them in our model, but it would be straightforward to add them. Such costs simply add another term to the well-behavedness condition, but do not change our results in any meaningful way.

While in this paper, we only model a single provider, some insights from our theoretical results also extend to *competitive* multi-provider settings. For such a setting, we would have to generalize our well-behavedness condition to multiple providers, which is straightforward. Following a similar argument as in the proof of Theorem 5.4, one could then show that, if none of the providers currently offers a spot market, then any provider for whom the well-behavedness condition is satisfied can increase her profit while simultaneously achieving a Pareto improvement for the users by offering a secondary spot market. Since this means that offering a spot market would be a profitable single-provider deviation from any strategy profile where none of the providers offer a spot market, there cannot be an equilibrium where no provider offers a spot market. We leave the formalization of multi-provider settings and a detailed study of the resulting equilibria to future work.

6 NUMERICAL EXAMPLES FOR MEMORYLESS QUEUES

In this section, we provide some numerical examples to illustrate the main results we have derived in the previous sections. So far, we have derived all theoretical results for *general* service processes. Now, we focus on fully *memoryless* queues, for which the well-known Erlang C formula (e.g., (Cooper 1981)) allows us to calculate queuing times and the expected number of preemptions

(given some additional technical assumptions). The formal model for the numerical examples and the corresponding formulas for the waiting time are provided in Appendix B.

6.1 Set-up

In our examples, we consider a setting with two classes of jobs. The parameters of the examples are as follows: the values for completion are $v = (1, 0.75)$, the arrival rates are $\lambda = (100, 50)$, and the expected service time is $\frac{1}{\mu} = 1$. The waiting costs c are uniformly distributed on $[0, 1]$ and $[0, 0.75]$, respectively. The SLA on the expected total waiting time for the fixed-price market is set to $T = 0.001$. For any job that joins the spot market, the expected number of external preemptions per time unit is $\psi_E(l_S) = l_S/100$. We assume that the provider can at most offer $l = 100$ spot instances; however, in our examples, this limit is never reached by the provider's optimal strategy. We set the preemption costs to $\tau = 0.25$ (i.e., whenever a job gets preempted it incurs additional running time equal to 25% of its "normal" service time $\frac{1}{\mu}$). We choose these relatively high costs to demonstrate that sizable profit increases are possible even with relatively costly preemptions.

6.2 Example 1: Varying the Instance Costs $\kappa = \kappa_F + \kappa_L$

For the first example, we vary the provider's instances costs $\kappa = \kappa_F + \kappa_L$ between $\kappa = 0$ and $\kappa = 0.2$. We assume that 90% of the costs are fixed-costs κ_F, while the remainder are load-dependent costs κ_L, i.e., $\kappa_F = 9\kappa_L$.

In Figure 1, we show the profit for different strategies, varying the instance costs κ on the x-axis. We plot the following four strategies. First, the red solid line shows the provider's profit-optimal strategy (denoted "hybrid"). Second, the dotted black line shows the profit-optimal strategy for when the provider is restricted to use the price p_F^{*0} that is optimal when only offering a fixed-price market (denoted "hybrid with $p_F = p_F^{*0}$"). Note that, by construction, this strategy guarantees a Pareto improvement for the users compared to the optimal fixed-price strategy. This is true because the users have access to the same fixed-price market, but additionally now also have access to a spot market (which some users prefer). Third, the dashed blue line shows the profit-optimal strategy for when the provider is restricted to only offering a fixed-price market (denoted "fixed-price"). Fourth, and as a reference, the dash-dotted green line shows the profit-optimal strategy for when no fixed-price market exists (denoted "spot-only").

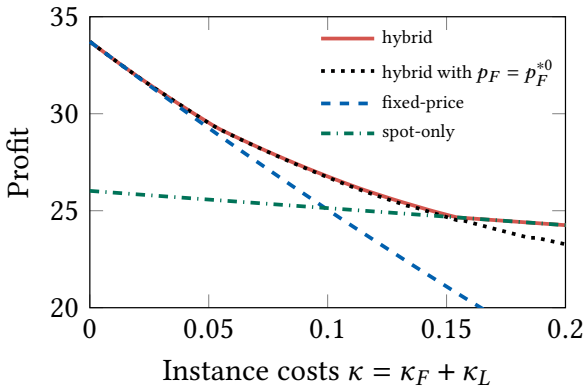

Fig. 1 Profit under different strategies while varying the instance cost κ

Looking at Figure 1, as the instance costs κ increase, we see the expected monotonic decrease of the profit for each of the four strategies. Note that, although the ratio between fixed and load-dependent costs stays constant, the profit of the fixed-price strategy decreases faster than for the other three strategies; the intuition for this is that the ratio between costs and revenue is highest in the fixed-price market. Importantly, we see that the hybrid strategy always achieves the highest profit among all four strategies which illustrates the main point of our paper. For very low

instance costs, its profit almost coincides with the profit of the fixed-price strategy. But already for moderate instance costs, the hybrid strategy leads to a significant profit increase. We also see that, when costs are very high, then the provider's optimal strategy is to set the price p_F high enough such that no user joins the fixed-price market (i.e., the equilibrium degenerates into a pure spot equilibrium).[17] Finally, we see that the hybrid strategy with $p_F = p_F^{*0}$ obtains almost the same profit as the (non-restricted) hybrid strategy (until the point where the hybrid strategy stops offering a fixed-price market). This illustrates that the provider does not need to compute a new price for her fixed-price market to achieve a sizeable profit increase via a hybrid strategy.

Overall, Example 1 illustrates our main result (Theorem 5.4): there exists a hybrid strategy (e.g., the hybrid strategy with $p_F = p_F^{*0}$)which simultaneously increases the provider's profit and leads to a Pareto improvement for the users.[18]

6.3 Example 2: Varying the Cost Ratio $\frac{\kappa_F}{\kappa_L}$

For the second example, we vary the ratio between fixed costs κ_F and load-dependent costs κ_L while keeping the sum constant at $\kappa = 0.1$. We again compare the same four strategies as in Example 1.

Figure 2 shows the profit for each strategy, varying the ratio $\frac{\kappa_F}{\kappa_L}$ between 0 and 20 on the x-axis. Thus, at 0, all costs are load-dependent, while at 20, the fixed costs are 20 times as high as the load-dependent costs. As we can see in Figure 2, at $\frac{\kappa_F}{\kappa_L} = 0$, offering no spot instances is optimal. This is expected because at 0, all costs are load-dependent, and therefore the main benefit of using idle instances (i.e., reducing fixed costs) is gone. As the fixed costs increase, the profits of the top three strategies (which at this point offer mostly fixed-price instances) first sharply decrease, even though the instance costs κ stay constant. This happens because the fixed-price instances now also incur a fraction of the instance costs while standing idle, whereas at 0, they only incurred costs while running. Conversely, spot instances get more attractive as $\frac{\kappa_F}{\kappa_L}$ increases. The hybrid strategies therefore start using spot instances to counteract the increased costs for offering fixed-price instances, which can be seen by the flattening of the solid red and dotted black lines.

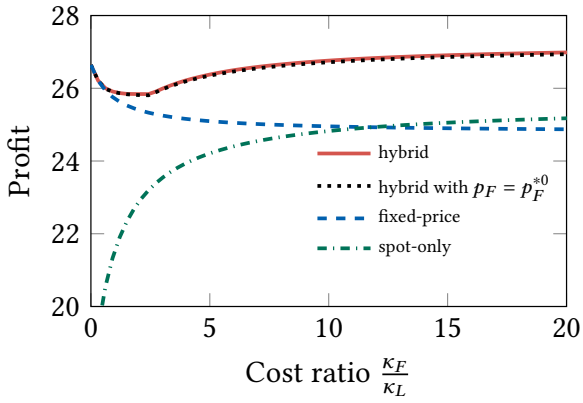

Fig. 2 Profit under different strategies while varying the cost ratio $\frac{\kappa_F}{\kappa_L}$

At the point where the fixed costs are about 3 times as large as the load-dependent costs, increasing $\frac{\kappa_F}{\kappa_L}$ further leads to a profit increase for both hybrid strategies. This happens because, beyond this point, both strategies offer enough spot instances such that the profit increase of the spot market dominates the profit decrease of the fixed-price market. We also again see that the hybrid strategy with $p_F = p_F^{*0}$ achieves close to optimal profits.

[17] Note that this result does not imply that in practice, the provider would not offer a fixed-price market, as there are always some users who would never join the spot market for a variety of reasons (e.g., because their jobs should never be preempted). Instead, this result shows that the provider's optimal strategy incentivizes all users who are *willing to consider* joining a spot market to do so (by setting a relatively high price).

[18] Note that a Pareto improvement for the users obviously implies a welfare increase. We demonstrate this welfare increase in Appendix C.

Recall that, in practice, fixed costs are usually about 10-20 times as large as load-dependent costs (i.e., $10 \leq \frac{\kappa_F}{\kappa_L} \leq 20$). Thus, this example suggests that, even when the provider incurs relatively large costs for spot instances (e.g., $\frac{\kappa_F}{\kappa_L} = 3$), a cloud provider can expect to achieve a sizable profit increase from offering a spot market in addition to her existing fixed-price market.

6.4 Varying the Provider Strategy

For the next two examples, we fix the fixed costs at $\kappa_F = 0.09$ and the load-dependent costs at $\kappa_L = 0.01$. We show how the profit changes, depending on different provider strategies $\rho = (p_F, l_S)$.

Example 3: Varying the Price p_F.
Figure 3 shows the optimal profit under two strategies that are restricted to the price p_F shown on the x-axis. The hybrid strategy (solid red line) offers the optimal number of spot instances l_S given p_F. The fixed-price strategy (dashed blue line) only offers a fixed-price market with price p_F.

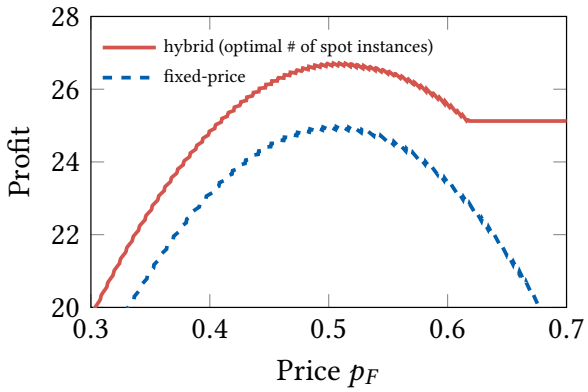

Fig. 3 Profit under different strategies when restricted to different prices p_F

Recall that the provider's profit depends on three factors: how many users join each market, how much they pay, and the provider's costs. As we can see in Figure 3, when the price p_F increases, the profit for both strategies at first also increases. This happens because the users' average payments under both strategies go up; this is also true in the spot market because increasing p_F pushes users from the fixed-price market into the spot market, which increases payments. However, since more users balk when the price increases, the profit only goes up until the rise in payments is dominated by the loss of users, whereafter the profit starts to fall again. Once the fixed-price market becomes too expensive for the users, the user equilibrium under the hybrid strategy degenerates into a pure spot equilibrium, as can be seen by the solid red line becoming constant beyond $p_F = 0.62$.

Example 4: Varying the Number of Spot Instances l_S.
Figure 4 shows the optimal profit under three strategies that are all restricted to offering the number of spot instances l_S shown on the x-axis. The hybrid strategy (solid red line) uses the optimal price p_F given l_S. The hybrid strategy with $p_F = p_F^{*0}$ (dotted black line) also offers l_S spot instances but uses the price that would be optimal when only offering a fixed-price market. As a reference, we also include the spot-only strategy (dash-dotted green line) which only offers a spot market with l_S spot instances.

As we see in Figure 4, as the number of spot instances l_S increases, the profit for the two hybrid strategies at first also increases. The key intuition for this is that, even though the average payments in the spot market may decrease, now more users join the spot market, where (on average) they

generate a higher profit than they previously did (either in the fixed-price market, or by balking). For both strategies, the optimal profit is achieved at $l_S = 39$, whereafter the profit starts to decrease.[19] Note that at $l_S = 62$, we again observe a point where the user equilibrium under the hybrid strategy degenerates into a pure spot equilibrium.

Finally, Figure 4 shows that for all numbers of spot instances, the profit achieved by the hybrid strategy with $p_F = p_F^{*0}$ is relatively close to the profit achieved with the optimal price. This suggests that, even when the provider cannot fully optimize her strategy, keeping the price at $p_F = p_F^{*0}$ is a viable approach for a cloud provider when offering a spot market.

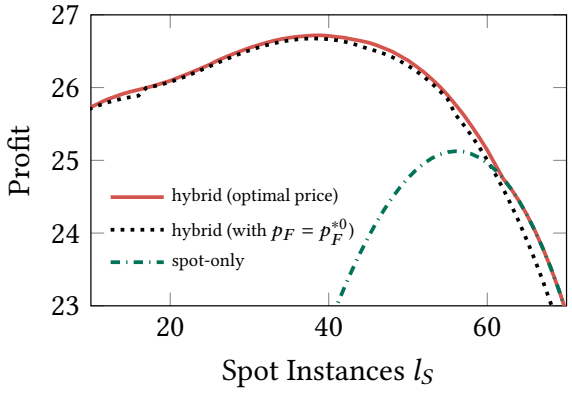

Fig. 4 Profit under different strategies when restricted to different spot market sizes l_S

7 CONCLUSION

In this paper, we have studied whether a cloud provider can benefit from selling idle instances on a spot market. Our main result is an easy-to-check condition under which a cloud provider can simultaneously increase her profit and achieve a Pareto increase for the users by offering a spot market in addition to a fixed-price market.

In contrast to prior work, we have modeled the provider's fixed and load-dependent costs as well as the users' costs for preemption. As these costs are an important factor for the profitability of any cloud market, modeling them was essential to make valid statements about the provider's profit optimization problem.

Our results have significant implications for practical market design. They suggest that, when our condition is satisfied, offering a spot market alongside her fixed-price market is advantageous for a cloud provider. Our condition is relatively mild and should be satisfied for most providers. Furthermore, even when a provider cannot compute her profit-optimal strategy, there are viable approaches to still achieve a profit increase by offering a spot market. Considering that the preemption costs are one of the main factors determining the profitability of a spot market, we encourage providers to continue to evolve their technology such that the losses incurred from re-starting a job are further reduced.

An interesting direction for future work would be to study how selling idle instances on a spot market compares to alternative market designs, such as the provider selling her idle instances on a preemptible *fixed-price* market. It is not immediately clear whether such a preemptible fixed-price market would be able to generate more or less profit than a spot market (with a reserve price). One would have to account for the differences in average payments, the market cannibalization towards the fixed-price market, and the costs produced by preemptions. Further, possible competitive advantages of one market over the other (i.e., differences in user satisfaction) would have to be taken into account. A complete analysis of this trade-off would be very valuable.

[19] This profit decrease is caused by three factors becoming dominant: the reduction in average payments, the need for a relatively larger buffer in the fixed-price market, and the unreliability of additional spot instances.

REFERENCES

Abhishek V, Kash IA, Key P (2012) Fixed and market pricing for cloud services. *2012 Proceedings IEEE INFOCOM Workshops*, 157–162.

Abhishek V, Kash IA, Key P (2017) Fixed and market pricing for cloud services, CoRR abs/1201.5621. Extended version of Abhishek et al. (2012).

Afèche P (2013) Incentive-compatible revenue management in queueing systems: Optimal strategic delay. *Manufacturing & Service Operations Management* 15(3):423–443.

Agmon Ben-Yehuda O, Ben-Yehuda M, Schuster A, Tsafrir D (2013) Deconstructing amazon ec2 spot instance pricing. *ACM Transactions on Economics and Computation* 1(3):16:1–16:20.

Azar Y, Kalp-Shaltiel I, Lucier B, Menache I, Naor J, Yaniv J (2015) Truthful online scheduling with commitments. *Proceedings of the 16th ACM Conference on Economics and Computation*, 715–732.

Babaioff M, Mansour Y, Nisan N, Noti G, Curino C, Ganapathy N, Menache I, Reingold O, Tennenholtz M, Timnat E (2017) Era: a framework for economic resource allocation for the cloud. *Proceedings of the 26th International Conference on World Wide Web Companion*, 635–642.

Banerjee S, Riquelme C, Johari R (2015) Pricing in ride-share platforms: A queueing-theoretic approach. *Proceedings of the 16th ACM Conference on Economics and Computation* 639.

Barroso LA, Hölzle U, Ranganathan P (2018) The datacenter as a computer: Designing warehouse-scale machines, third edition. *Synthesis Lectures on Computer Architecture* 13(3):i–189.

Boodaghians S, Fusco F, Leonardi S, Mansour Y, Mehta R (2019) Online revenue maximization for server pricing. *CoRR* abs/1906.09880.

Buzen JP, Bondi AB (1983) The response times of priority classes under preemptive resume in m/m/m queues. *Operations Research* 31(3):456–465.

Cohen MC, Keller PW, Mirrokni V, Zadimoghaddam M (2019) Overcommitment in cloud services: Bin packing with chance constraints. *Management Science* 65(7):3255–3271.

Cooper RB (1981) *Introduction to queueing theory* (Amsterdam, NL: North Holland Publishing Co.).

Cortez E, Bonde A, Muzio A, Russinovich M, Fontoura M, Bianchini R (2017) Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms. *Proceedings of the 26th Symposium on Operating Systems Principles*, 153–167.

Desai PS (2001) Quality segmentation in spatial markets: When does cannibalization affect product line design? *Marketing Science* 20(3):265–283.

Dierks L, Kash IA, Seuken S (2019) On the cluster admission problem for cloud computing. *Proceedings of the 14th Workshop on the Economics of Networks, Systems and Computation.*

Dierks L, Seuken S (2019) Cloud pricing: The spot market strikes back (extended abstract). *Proceedings of the 20th ACM Conference on Economics and Computation*, 593.

Dierks L, Seuken S (2020) The competitive effects of variance-based pricing. *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, Forthcoming.

Gao J, Iyer K, Topaloglu H (2019) When fixed price meets priority auctions: Competing firms with different pricing and service rules. *Stochastic Systems* 9(1):47–80.

Hassin R (2016) *Rational Queueing* (Boca Raton, FL: CRC Press).

Hassin R, Haviv M (2003) *To queue or not to queue: Equilibrium behavior in queueing systems* (Norwell, MA: Kluwer Academic Publishers).

Hoy D, Immorlica N, Lucier B (2016) On-demand or spot? selling the cloud to risk-averse customers. *International Conference on Web and Internet Economics*, 73–86.

Islam M, Ren X, Ren S, Wierman A (2018) A spot capacity market to increase power infrastructure utilization in multi-tenant data centers. *2018 IEEE International Symposium on High Performance Computer Architecture*, 776–788.

Jyothi SA, Curino C, Menache I, Narayanamurthy SM, Tumanov A, Yaniv J, Mavlyutov R, Goiri I, Krishnan S, Kulkarni J, et al. (2016) Morpheus: Towards automated slos for enterprise clusters. *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, 117–134.

Kash IA, Key P (2016) Pricing the cloud. *IEEE Internet Computing* 20(1):36–43.

Kash IA, Key P, Suksompong W (2017) Simple pricing schemes for the cloud. *International Conference on Web and Internet Economics*, 311–324.

Maharjan S, Zhu Q, Zhang Y, Gjessing S, Basar T (2013) Dependable demand response management in the smart grid: A stackelberg game approach. *IEEE Transactions on Smart Grid* 4(1):120–132.

Maskin E, Riley J (1984) Monopoly with incomplete information. *The RAND Journal of Economics* 15(2):171–196.

Moorthy KS (1984) Market segmentation, self-selection, and product line design. *Marketing Science* 3(4):288–307.

Mussa M, Rosen S (1978) Monopoly and product quality. *Journal of Economic Theory* 18(2):301 – 317.

Myerson RB (1981) Optimal auction design. *Mathematics of operations research* 6(1):58–73.

Shaked A, Sutton J (1982) Relaxing price competition through product differentiation. *The review of economic studies* 3–13.

Shi W, Zhang L, Wu C, Li Z, Lau FC (2014) An online auction framework for dynamic resource provisioning in cloud computing. *The 2014 ACM International Conference on Measurement and Modeling of Computer Systems*, 71–83.

Subramanya S, Rizk A, Irwin D (2016) Cloud spot markets are not sustainable: The case for transient guarantees. *8th USENIX Workshop on Hot Topics in Cloud Computing*.

Takagi H (2008) *Spectrum Requirement Planning in Wireless Communications*, chapter Appendix A: Derivation of Formulas by Queueing Theory, 199–218 (John Wiley & Sons, Ltd).

Varian HR (1989) Price discrimination. *Handbook of industrial organization* 1:597–654.

Wolff RW (1982) Poisson arrivals see time averages. *Operations Research* 30(2):223–231.

Yan Y, Gao Y, Chen Y, Guo Z, Chen B, Moscibroda T (2016) Tr-spark: Transient computing for big data analytics. *Proceedings of the 7th ACM Symposium on Cloud Computing*, 484–496.

Zaharia M, Borthakur D, Sen Sarma J, Elmeleegy K, Shenker S, Stoica I (2010) Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. *Proceedings of the 5th European conference on Computer systems*, 265–278.

Zhang H, Jiang H, Li B, Liu F, Vasilakos AV, Liu J (2016) A framework for truthful online auctions in cloud computing with heterogeneous user demands. *IEEE Transactions on Computers* 65(3):805–818.

Zhang L, Li Z, Wu C (2014) Dynamic resource provisioning in cloud computing: A randomized auction approach. *2014 IEEE Conference on Computer Communications*, 433–441.

Zheng L, Joe-Wong C, Brinton CG, Tan CW, Ha S, Chiang M (2016) On the viability of a cloud virtual service provider. *ACM SIGMETRICS Performance Evaluation Review* 44(1):235–248.

Zhou R, Li Z, Wu C, Huang Z (2017) An efficient cloud market mechanism for computing jobs with soft deadlines. *IEEE/ACM Transactions on Networking* 25(2):793–805.

A ADDITIONAL STATEMENTS AND PROOFS

A.1 Lemma A.1

A number of our results make use of the following technical Lemma which extends Lemma 7 from Abhishek et al. (2017) to more general g_i and $w()$, which includes our model.

Lemma A.1. *Fix a provider strategy ρ with $l_s > 0$. Let $(x_1, \dots, x_k)$ be a weakly decreasing sequence. For $i > k$ let $g_i(x_i), \dots, g_n(x_n)$ be given such that $g_j(x_j)$ is a weakly increasing and semi-differentiable scalar function with left derivative at most $w(\mathcal{S}, x_i, \rho, (x_1, \dots, x_k, x_i \dots, x_i, 0_{i+1}, \dots, 0_n))$. (Here, the notation $(x_1, \dots, x_k, x_i \dots, x_i, 0_{i+1}, \dots, 0_n)$ means that all entries k' with $k < k' \leq i$ are set to x_i.) Further, assume that for all $j \geq i$ it holds that $g_j(\mu v_j) \leq v_j$, as well as $g_j(x) \leq g_i(x)$ for all $x \in \mathbb{R}$. Then there exist unique $x_i, \dots, x_n$ such that for any strategy profile*

$$\sigma' = \overrightarrow{x} \quad = \quad (x_1, \dots, x_k, x_i, \dots, x_i, x_{i+1}, \dots, x_n) \tag{50}$$

where any user of class j joins the spot market if and only if his waiting cost c is below x_j, it holds that

$$\int_0^{x_j} w(\mathcal{S}, c, \rho, \overrightarrow{x}) dc \quad = \quad g_j(x) \quad \text{for all } j \geq i. \tag{51}$$

Proof. To see this, assume that the claim holds for $i+1$. Then for any $z \in [0, x_k]$ there exists

$$\sigma(z) = \overrightarrow{x}(z) \quad = \quad (x_1, \dots, x_k, z, \dots, z, x_{i+1}(z), \dots, x_n(z)) \tag{52}$$

satisfying Equation (51) for any $j \geq i+1$. We now show that there exists a unique z^* such that $w(z^*) = \int_0^{z^*} w(\mathcal{S}, c, \rho, \sigma(z^*)) dz$. As a first step, we show that $w(z) = \int_0^z w(\mathcal{S}, c, \rho, \sigma(z)) dz$ is increasing in z. Since for any fixed x, $\int_0^x w(\mathcal{S}, c, \rho, \overrightarrow{x}) dc$ is increasing in each x_j, it follows that $x_{i+1}(z)$ is decreasing in z. Then for any $\hat{z} > z$ it holds by the induction assumption that

$$\int_0^{x_{i+1}(z)} w(\mathcal{S}, c, \rho, \overrightarrow{x}(\hat{z})) dc \quad \geq \quad g_i(x_{i+1}(z)) \tag{53}$$

and therefore

$$\frac{\int_0^{\hat{z}} w(\mathcal{S}, c, \rho, \overrightarrow{x}(\hat{z})) dc - \int_0^z w(\mathcal{S}, c, \rho, \overrightarrow{x}(z)) dc}{\hat{z} - z} \tag{54}$$

$$\geq \quad \frac{g_{i+1}(x_{i+1}(z)) - g_{i+1}(x_{i+1}(z))}{\hat{z} - z} + \frac{\int_{x_{i+1}(z)}^{\hat{z}} w(\mathcal{S}, c, \rho, \overrightarrow{x}(\hat{z})) dc - \int_{x_{i+1}(z)}^z w(\mathcal{S}, c, \rho, \overrightarrow{x}(z)) dc}{\hat{z} - z} \tag{55}$$

$$= \quad \frac{\int_z^{\hat{z}} w(\mathcal{S}, c, \rho, \overrightarrow{x}(\hat{z})) dc + \int_{x_{i+1}(z)}^z w(\mathcal{S}, c, \rho, \overrightarrow{x}(\hat{z})) - w(\mathcal{S}, c, \rho, \overrightarrow{x}(z)) dc}{\hat{z} - z} \tag{56}$$

$$> \quad \frac{\int_z^{\hat{z}} w(\mathcal{S}, c, \rho, \overrightarrow{x}(\hat{z})) dc}{\hat{z} - z} \tag{57}$$

$$> \quad w(\mathcal{S}, z, \rho, \overrightarrow{x}(\hat{z})). \tag{58}$$

Equation (57) is a direct result of the fact that waiting times of higher priority jobs do not depend on the number of lower priority jobs in the queue. By taking the limit $\hat{z} \to z$ it follows that $w'(z) > w(\mathcal{S}, z, \rho, \overrightarrow{x}(\hat{z})) \geq w(\mathcal{S}, z, \rho, (x_1, \dots, x_k, z \dots, z, 0_{i+1}, \dots, 0_n)$. As $w(0) = 0$ and $w(\mu v_i) \geq v_i \geq g_i(\mu v_i)$, the claim for i follows.

To show the induction base case $i = n$, we introduce a dummy variable x_{n+1} with $g_{n+1} = 0$. This means that for any z it trivially holds that $x_{n+1}(z) = 0$ and the statement for n follows. □

A.2 Proof of Proposition 4.2

Any equilibrium strategy for users of class i is trivially a threshold strategy because for any fixed strategy profile σ, the payoff of a user of class i, i.e., $\pi_i^c(\mathcal{S}, c, \rho, \sigma)$, is monotone decreasing in the waiting cost c. By setting $g_i(c) = v_i$, the existence of the unique cutoff vector $\overrightarrow{c}^S$ then follows directly from Lemma A.1. By the incentive compatibility of the payment rule, no user would deviate from $\sigma^* = \overrightarrow{c}^S$.

A.3 Proof of Proposition 4.3

Users with waiting close enough to zero always prefer the spot market, no matter how much time they lose compared to the fixed-price market. Since some users join the fixed-price market and both waiting time and payment are continuous in the bid c, for any potential equilibrium strategy profile σ^*, there has to be a lowest point c_1^P such that

$$c_1^P(T + \frac{1}{\mu}) + p_F\frac{1}{\mu} = \int_0^{c_1^P} w(\mathcal{S}, x, \rho, \sigma)dx. \tag{59}$$

Since it holds $\frac{d}{dc}\pi_i^c(\mathcal{S}, c, \rho, \sigma) = w(\mathcal{S}, c, \rho, \sigma) \geq \frac{1}{\mu}\frac{1}{1-\tau\psi_E(l_S)} > T + \frac{1}{\mu} = \frac{d}{dc}\pi_i^c(\mathcal{F}, c, \rho, \sigma)$, the higher a users waiting cost, the worse the spot market compared to the fixed-price market and there cannot exist any $c > c_1^P$ for which users prefer the spot market, i.e. with

$$c(T + \frac{1}{\mu}) + p_F\frac{1}{\mu} > \int_0^{c} w(\mathcal{S}, x, \rho, \sigma)dx. \tag{60}$$

Thus, no user with waiting cost greater than c_1^P joins the spot market. This means that the spot market can be fully defined by the actions of players with $c \leq c_1^P$. Recall that $\overrightarrow{c}^P$ denotes the vector of cutoff points at which a job becomes indifferent between the spot market and either the fixed-price market or balking. It holds that

$$\min\left\{\frac{p_F}{\mu} + c_i^P(\frac{1}{\mu} + T), v_i\right\} \quad = \quad \int_0^{c_i^P} w(\mathcal{S}, x, \rho, \overrightarrow{c}^P)dx \quad \forall i \in \{1, \ldots, n\}, \tag{61}$$

which has a unique solution by Lemma A.1. Every job of class i with $c < c_i^P$ joins the spot market, and every job with $c_i^P < c < \frac{\mu v_i - p_F}{\mu T+1}$ joins the fixed-price market and those with $c > \frac{\mu v_i - p_F}{\mu T+1}$ balk. Setting $c_i^B = max(c_i^P, \frac{\mu v_i - p_F}{\mu T+1})$, it is clear that every solution of (17) and (18) solves (61) and vice-versa.

A.4 Lemma A.2

The following Lemma establishes the broad equilibrium structure when the spot market is faster at the highest bids and shows the existence of two cutoff points in equilibrium between which almost all users join the fixed-price market. It is used in the proof of Proposition 4.4.

Lemma A.2. *For any provider strategy $\rho = (p_F, l_S)$, in any BNE of the user game where some users join the fixed-price market and where the spot market is faster for the highest bids, i.e., $T > \frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right)$, there exists an interval $[c^L, c^U]$, such that almost all users (i.e., all besides possibly a set of measure zero that does not influence system dynamics) with waiting costs $c \in [c^L, c^U]$ join the fixed-price market or balk. For bids $c \in [c^L, c^U]$, the total waiting time and the payment in equilibrium are*

equal in each market, i.e.:

$$m(\mathcal{S}, c, \rho, \sigma^*) = m(\mathcal{F}, c, \rho, \sigma^*) = p_F \frac{1}{\mu} \tag{62}$$

$$w(\mathcal{S}, c, \rho, \sigma^*) = w(\mathcal{F}, c, \rho, \sigma^*) = T + \frac{1}{\mu} \tag{63}$$

For waiting costs $c \notin [c^L, c^U]$ it holds that

$$\pi_i(\mathcal{S}, c, \rho, \sigma^*) > \pi_i(\mathcal{F}, c, \rho, \sigma^*) \;\; \forall i \in \{1, \ldots, n\}, \tag{64}$$

and these users join the spot market or balk.

Proof. For a job with the highest bid that does not balk, the spot market is faster than the fixed-price market because $T > \frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right)$. Any user with such a waiting cost is therefore willing to pay more in the spot market than in the fixed-price market. This means that he strictly prefers the spot market in equilibrium.

Let c^L be the lowest waiting cost for which a job prefers the fixed-price market over the spot market or is indifferent between the two, and let c^U be the highest such waiting cost. c^L and c^U have to exist for any equilibrium where users join both markets. We now show by contradiction that the user's payment in the spot market has to be weakly larger than in the fixed-price market for bids above c^L and that the spot market is weakly slower than the fixed-price market for bids below c^U.

Assume there exists a waiting cost $\bar{c} > c^L$ at which a user would prefer the spot market or be indifferent between spot and fixed-price market, and for which the payment in the spot market is less in expectation than in the fixed-price market, i.e., for which $m(\mathcal{S}, \bar{c}, \rho, \sigma) < m(\mathcal{F}, \bar{c}, \rho, \sigma)$. Then

$$\begin{aligned}
& c^L w(\mathcal{S}, c^L, \rho, \sigma) + m(\mathcal{S}, c^L, \rho, \sigma) && (65)\\
\leq\; & c^L w(\mathcal{S}, \bar{c}, \rho, \sigma) + m(\mathcal{S}, \bar{c}, \rho, \sigma) && (66)\\
=\; & \frac{c^L}{\bar{c}}\left(\bar{c} w(\mathcal{S}, \bar{c}, \rho, \sigma) + \frac{\bar{c}}{c^L} m(\mathcal{S}, \bar{c}, \rho, \sigma)\right) && (67)\\
\leq\; & \frac{c^L}{\bar{c}}\left(\bar{c} w(\mathcal{F}, \bar{c}, \rho, \sigma) + m(\mathcal{F}, \bar{c}, \rho, \sigma) + (\frac{\bar{c}}{c^L} - 1) m(\mathcal{S}, \bar{c}, \rho, \sigma)\right) && (68)\\
<\; & \frac{c^L}{\bar{c}}\left(\bar{c} w(\mathcal{F}, \bar{c}, \rho, \sigma) + m(\mathcal{F}, \bar{c}, \rho, \sigma) + (\frac{\bar{c}}{c^L} - 1) m(\mathcal{F}, \bar{c}, \rho, \sigma)\right) && (69)\\
=\; & \frac{c^L}{\bar{c}}\left(\bar{c} w(\mathcal{F}, \bar{c}, \rho, \sigma) + \frac{\bar{c}}{c^L} m(\mathcal{F}, \bar{c}, \rho, \sigma)\right) && (70)\\
=\; & c^L w(\mathcal{F}, \bar{c}, \rho, \sigma) + m(\mathcal{F}, \bar{c}, \rho, \sigma) && (71)\\
=\; & c^L w(\mathcal{F}, c^L, \rho, \sigma) + m(\mathcal{F}, c^L, \rho, \sigma) && (72)
\end{aligned}$$

(66) holds because the pricing rule is BNIC; (68) holds because at waiting cost $\bar{c}$ the spot market's overall cost has to be lower than the fixed-price market in order for the user to join it. Finally, (69) holds because we assumed the spot market to be cheaper with bid $\bar{c} > c^L$. A job with waiting cost c^L would therefore also strictly prefer the spot market, a contradiction.

Assume there exists a waiting cost $\bar{c} < c^U$ at which a user would prefer the spot market or be indifferent between spot and fixed-price market, and for which the waiting time is lower in the spot

market than in the fixed-price market, i.e., for which $w(\mathcal{S}, \bar{c}, \rho, \sigma) < w(\mathcal{F}, \bar{c}, \rho, \sigma)$. Then similarly

$$
\begin{aligned}
& c^U w(\mathcal{S}, c^U, \rho, \sigma) + m(\mathcal{S}, c^U, \rho, \sigma) && (73)\\
\leq\ & c^U w(\mathcal{S}, \bar{c}, \rho, \sigma) + m(\mathcal{S}, \bar{c}, \rho, \sigma) && (74)\\
=\ & \bar{c} w(\mathcal{S}, \bar{c}, \rho, \sigma) + m(\mathcal{S}, \bar{c}, \rho, \sigma) + (c^U - \bar{c}) w(\mathcal{S}, \bar{c}, \rho, \sigma) && (75)\\
<\ & \bar{c} w(\mathcal{F}, \bar{c}, \rho, \sigma) + m(\mathcal{F}, \bar{c}, \rho, \sigma) + (c^U - \bar{c}) w(\mathcal{F}, \bar{c}, \rho, \sigma) && (76)\\
=\ & c^U w(\mathcal{F}, \bar{c}, \rho, \sigma) + m(\mathcal{F}, \bar{c}, \rho, \sigma) && (77)
\end{aligned}
$$

A user with c^U would therefore also strictly prefer the spot market, a contradiction.

Therefore, for all $c \in [c^L, c^U]$ the spot market can neither be faster nor cheaper than the fixed-price market. If any users with waiting cost $c \in [c^L, c^U]$ join the spot market they have to be indifferent between both markets. Thus, for any σ^* to be a BNE, this means that at most a set of measure zero of such users can join the spot market, and thus the total waiting time and payment stay constant over the whole interval. The statement of the lemma immediately follows. □

A.5 Proof of Proposition 4.4

It follows from Lemma A.2 that any equilibrium strategy profile is of the form $\sigma^* = (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$: Let the points c^L and c^U be as given by Lemma A.2 and let $\overrightarrow{c}^H$ denote the waiting costs above which users of each class cannot obtain a positive payoff anymore and balk. Define the cutoff vectors $\overrightarrow{c}^L, \overrightarrow{c}^U$ as $c_i^L = \min\left\{c^L, c_i^H\right\}$ and $c_i^U = \min\left\{c^U, c_i^H\right\}$. Note that this implies that $c_1^L = c^L$ and $c_1^U = c^U$ (because at least some users from class 1 go to the portion of the spot market that is faster than the fixed-price market). Then Equations (19), (20) and (21) immediately follow from Lemma A.2:

(1) Equation (19): The payoff at c^L has to be the same for joining the fixed-price or spot market.
(2) Equation (20): The payoff at c^U also has to be the same in the fixed-price and spot market.
(3) Equation (21): Users do not balk as long as their value for joining one of the markets is greater than 0.

We now show that this system of equations always has a unique solution using a constructive approach. For this, we first introduce some additional notation.

Given provider strategy ρ, we know that in order to satisfy Lemma A.2, jobs that pay more in the spot market than in the fixed-price marked need to arrive at a rate such that jobs with waiting cost c^L have to queue for exactly T. Denote this arrival rate by $\lambda(T, \rho)$. We now further overload our previous notation for a user strategy profile: for any vector $\hat{c} = (\hat{c}_1, \ldots, \hat{c}_n)$ with $\hat{c}_i \geq \hat{c}_j$ for $i < j$, we let $\sigma = (\hat{c}, \lambda(T, \rho))$ denote a user strategy profile where every user of class i with waiting cost $c < \hat{c}_i$ joins the spot market, but everyone else balks even if he could obtain a positive payoff in one of the markets. Additionally, we assume that *dummy jobs* of maximal priority arrive with rate $\lambda(T, \rho)$ into the spot market. Thus, $w(\mathcal{S}, \hat{c}_1, \rho, (\hat{c}, \lambda(T, \rho))) = T$ by definition. Combined with Lemma A.2, this notational trick allows us to "simulate" the impact users with waiting costs between $\overrightarrow{c}^U$ and $\overrightarrow{c}^H$ have on all other users, without yet knowing $\overrightarrow{c}^U$ and $\overrightarrow{c}^H$. We now need to determine which classes of users join the fixed-price market (in the sense that there exists a c such that a user from that class with waiting cost c joins the fixed-price market) and which do not. Once we know that, we can split the system of equations into two parts that can be solved consecutively.

To check whether the k'th class joins the fixed-price market, i.e., whether $c_k^H > c^L$, we denote by $\hat{c} = (\hat{c}_1, \ldots, \hat{c}_n)$ (where each $\hat{c}_i \in [0, \mu v_i]$) the cutoff vector solving the following:

$$0 = v_i - \int_0^{\hat{c}_i} w(\mathcal{S}, x, \rho, (\hat{c}, \lambda(T, \rho)))dx \quad \forall i \geq k \tag{78}$$

$$\hat{c}_i = \hat{c}_k \quad \forall i < k \tag{79}$$

This has a unique solution according to Lemma A.1. Note that the cutoff vector $\hat{c}$ here carries an implicit dependence on k, while $\hat{c}_k$ denotes its k'th element.
If it now holds that

$$p_F \frac{1}{\mu} > m(\mathcal{S}, \hat{c}_k, \rho, (\hat{c}, \lambda(T, \rho))) \tag{80}$$

$$= \int_0^{\hat{c}_k} w(\mathcal{S}, x, \rho, (\hat{c}, \lambda(T, \rho)))dx - w(\mathcal{S}, \hat{c}_k, \rho, (\hat{c}, \lambda(T, \rho))), \tag{81}$$

then $c^L \leq \hat{c}_k$ would mean that not enough users join the spot market to reach the price of the fixed-price market at the cutoff point c^L. This means that for any such c^L, the system of equations defining the equilibrium cutoff vectors (i.e., Equations (19), (20) and (21)) cannot be satisfied for any choice of c^U and $\overrightarrow{c}^H$. It follows that in equilibrium, $c^L > \hat{c}_k$. Thus, in equilibrium, no user of class k joins the fixed-price market, i.e., $c_k^H < c^L$.

Conversely, if Equation (80) does not hold, then setting $c^L > \hat{c}_k$ would mean that too many users join the spot market, and the payment in the spot market at the cutoff point c^L is larger than the payment in the fixed-price market. Thus, in any equilibrium, some users of class k join the fixed-price market and $c^L \leq \hat{c}_k$. As $m(\mathcal{S}, \hat{c}_k, \rho, (\hat{c}, \lambda(T, \rho)))$ is monotone decreasing in k, it follows that either there exists a lowest class k^* such that (81) is satisfied and for which no user joins the fixed-price market, or all classes join the fixed-price market in which case we set $k^* = n + 1$. Splitting the system of equations that defines the equilibrium strategy profile at this k^*, Lemma A.1 yields that

$$0 = \hat{c}_i(T + \frac{1}{\mu}) + p_F \frac{1}{\mu} - \int_0^{\hat{c}_i} w(\mathcal{S}, x, \rho, (\hat{c}, \lambda(T, \rho)))dx \quad \forall i < k^* \tag{82}$$

$$0 = v_i - \int_0^{\hat{c}_i} w(\mathcal{S}, x, \rho, (\hat{c}, \lambda(T, \rho)))dx \quad \forall i \geq k^* \tag{83}$$

has a unique solution with $\hat{c}_{k^*} < \hat{c}_{k^*-1}$ and for any equilibrium $(\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$ it holds that $c^L = \hat{c}_{k^*-1}$ and $c_i^H = \hat{c}_i$ for all $i \geq k^*$.

Given the solution to (82) and (83), we can now equivalently find the highest class k^{**} that joins the upper portion of the spot market (i.e. for which $c_{k^{**}}^U < c_{k^{**}}^H$). To this end, fix any $k < k^*$. Again carrying an implicit dependence on k, we define temporary cutoff vectors $\hat{c}^U$ and $\hat{c}^H$. Set $\hat{c}_i^H = c^L$ for all $k < i < k^*$ and $\hat{c}_i^H = \overrightarrow{c}_i^H$ for all $i \geq k^*$. Further let $\hat{c}^U$ and $\hat{c}_i^H$ for $i \leq k$ be given as the solution to

$$0 = v_{k+1} - \hat{c}_{k+1}^U(T + \frac{1}{\mu}) + p_F \frac{1}{\mu} \tag{84}$$

$$0 = v_i - \int_0^{\hat{c}_i^H} w(\mathcal{S}, x, \rho, (\overrightarrow{c}^L, \hat{c}^U, \hat{c}^H))dx \quad \forall i \leq k \tag{85}$$

$$\hat{c}_i^U = min(\hat{c}_k^U, \hat{c}_i^H) \quad \forall i \neq k + 1. \tag{86}$$

This system of equations has a unique solution for every $k < k^*$ according to Lemma A.1. Intuitively, $(\overrightarrow{c}^L, \hat{c}^U, \hat{c}^H)$ can be seen as the strategy profile where users with waiting costs below c^L play the equilibrium strategy, no user joins the fixed-price market, and users with waiting costs above the point where class $k+1$ would obtain zero payoff in the fixed-price market join the spot market (if their payoff for doing so is positive). This means that under this strategy profile *more* users join the spot market than would under any potential equilibrium strategy profile where $c^U > \hat{c}^U_{k+1}$. Analogous to k^*, there now exists a lowest class $k^{**} < k^*$, such that if only jobs of classes $i \leq k^{**}$ join the upper part of the spot market, there are still enough users that potentially (i.e., as long as the fixed-price market isn't better) join the spot market, such that the waiting time in the spot market at $\hat{c}^U_k$ is at least as high as the waiting time in the fixed-price market, i.e., k^{**} is the smallest k for which it holds that

$$T + \frac{1}{\mu} \quad \leq \quad w(\mathcal{S}, \hat{c}^U_k, \rho, (\overrightarrow{c}^L, \hat{c}^U, \hat{c}^H)). \tag{87}$$

Conversely, if users of classes higher than k^{**} would join the upper portion of the spot market (i.e. $c^U \leq \hat{c}^U_{k^{**}+1}$) then the waiting time in the spot market at c^U is always above $T + \frac{1}{\mu}$. Consequently, we can calculate $\overrightarrow{c}^H_i$ for $k^{**} < i < k^*$ as the solution to

$$0 \quad = \quad v_i - \overrightarrow{c}^H_i (T + \frac{1}{\mu}) + p_F \frac{1}{\mu}. \tag{88}$$

Then finally, we can calculate c^U and $\overrightarrow{c}^H_i$ for $i \leq k^{**}$ as the solution to

$$0 \quad = \quad c^U (T + \frac{1}{\mu}) + p_F \frac{1}{\mu} - \int_0^{c^U} w(\mathcal{S}, x, \rho, (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)) dx \tag{89}$$

$$0 \quad = \quad v_i - \int_0^{c^H_i} w(\mathcal{S}, x, \rho, (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)) dx \quad \forall i \leq k^{**}, \tag{90}$$

which, given c^L and c^H_i for all $i > k^{**}$ now also has a unique solution according to Lemma A.1.

As each of the successively solved subsystems of equations was, at the time it was solved, independent of the then unsolved parts, $(\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$ solves the whole system of equations.

A.6 Proof of Lemma 4.6

For $l_S > 0$, all users with waiting costs in some neighborhood around zero prefer the spot market. Let ρ be a provider strategy with $l_S > 0$ that is proper. Assume we have an equilibrium where no one joins the fixed-price market, i.e., where the hybrid market degenerates to the spot market. A user of class 1 (i.e., the class with maximal value for completion) with waiting cost c^S_1 (if $\frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right) \geq T$) or c' (if $\frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right) < T$) could then obtain a better payoff by switching to the fixed-price market, leading to a contradiction. Any BNE therefore has some users joining the fixed-price market.

Now assume ρ is not proper. We first show the statement for $\frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right) \geq T$ (i.e., when the spot market is always slower than the fixed-price market). Proposition 4.3 gives us that any equilibrium where some users join the fixed-price market is of the form $\sigma = (\overrightarrow{c}^P, \overrightarrow{c}^B)$. At any waiting cost c for which users of some class i balk under $\sigma = (\overrightarrow{c}^S)$ but join the spot market under $\sigma = (\overrightarrow{c}^P, \overrightarrow{c}^B)$, their payoff in the spot market needs to be higher under $\sigma = (\overrightarrow{c}^P, \overrightarrow{c}^B)$, i.e.,

$$\pi^c_i(\mathcal{S}, c, \rho, (\overrightarrow{c}^P, \overrightarrow{c}^B)) \geq 0 \geq \pi^c_i(\mathcal{S}, c, \rho, (\overrightarrow{c}^S)). \tag{91}$$

The payment and waiting time in the spot market are only *changing* in the bid c at bids for which any users go into the spot market. It directly follows that a user's payoff in the spot market (if he

were to choose it) is (weakly) higher for *any* user (and thus also for users with waiting cost c_1^S) under $\sigma = (\overrightarrow{c}^S)$ than under $\sigma = (\overrightarrow{c}^P, \overrightarrow{c}^B)$. If ρ is not proper, it follows

$$\pi_1^{c_1^S}(\mathcal{S}, c_1^S, \rho, (\overrightarrow{c}^P, \overrightarrow{c}^B)) \geq \pi_1^{c_1^S}(\mathcal{S}, c_1^S, \rho, (\overrightarrow{c}^S)) > \pi_1^{c_1^S}(\mathcal{F}, c_1^S, \rho, (\overrightarrow{c}^S)) = \pi_1^{c_1^S}(\mathcal{F}, c_1^S, \rho, (\overrightarrow{c}^P, \overrightarrow{c}^B)), \tag{92}$$

i.e., users of class 1 with waiting cost c_1^S would deviate from $\sigma = (\overrightarrow{c}^P, \overrightarrow{c}^B)$, contradicting that $\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$ is a BNE.

Now we show the statement for when ρ is not proper and when $\frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right) < T$ (i.e., when the spot market is faster for the highest bids). If no c' exists for which the expected waiting time in both queues is equal (i.e., for which condition (a) from Definition 4.5 holds), the spot market is trivially faster for every user (and consequently also cheaper) and the statement follows by the same argument as for $\frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right) \geq T$.

Now assume there exists a c' satisfying condition (a), but it does not satisfy condition (b), i.e.

$$c'(T + \frac{1}{\mu}) + p\frac{1}{\mu} \quad \geq \quad \int_0^{c'} w(\mathcal{S}, x, \rho, \sigma)dx \tag{93}$$

for c' such that $T + \frac{1}{\mu} = w(\mathcal{S}, c', \rho, \sigma)$. It follows that

$$p_F\frac{1}{\mu} \quad \geq \quad \int_0^{c'} w(\mathcal{S}, x, \rho, \sigma)dx - c'w(\mathcal{S}, c', \rho, \sigma). \tag{94}$$

This means that even with bid c', for which the fixed-price market has the same total waiting time as the spot market, joining the spot market is still cheaper. We now show that, in this case, there only exist BNEs where no user joins the fixed-price market. Since the payoff in the spot market for every user is monotone decreasing in the number of users that join, it is enough to show that when playing $\sigma = \overrightarrow{c}^S$, no user has an incentive to switch to the fixed-price market.

A user with waiting cost c' clearly has no reason to switch. Assume that a user of class i with waiting cost $c \neq c'$ would prefer to switch to the fixed-price market. Misreporting his class as c' and joining the spot market would then lead to a payoff of

$$\begin{aligned}
\pi_i^c(\mathcal{S}, c', \rho, \overrightarrow{c}^S) &= v_i - \int_0^{c'} w(\mathcal{S}, x, \rho, \sigma)dx + c'w(\mathcal{S}, c', \rho, \sigma) - cw(\mathcal{S}, c', \rho, \sigma) && (95)\\
&= v_i - \int_0^{c'} w(\mathcal{S}, x, \rho, \sigma)dx + c'w(\mathcal{S}, c', \rho, \sigma) - c(T + \frac{1}{\mu}) && (96)\\
&\geq v_i - p_F\frac{1}{\mu} - c(T + \frac{1}{\mu}) && (97)\\
&= \pi_i^c(\mathcal{F}, c, \rho, \overrightarrow{c}^S) && (98)\\
&> \pi_i^c(\mathcal{S}, c, \rho, \overrightarrow{c}^S) && (99)
\end{aligned}$$

Misreporting in the spot market would therefore be beneficial over reporting truthfully, contradicting the pricing rule being Bayes-Nash incentive compatible. Consequently, no user prefers the fixed-price market and by Theorem 4.2 it holds that $\sigma^* = \overrightarrow{c}^S$.

A.7 Lemma A.3

The proof of Theorem 5.4 requires the introduction of an additional technical Lemma. The following Lemma establishes that the average payments in the spot market approach the payments in the fixed-price market for small enough l_S.

Lemma A.3. *For any strategy $\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$ or $\sigma^* = (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$, denote by $\bar{m}(\mathcal{S}, \rho, \sigma^*)$ the average payment in the spot market, i.e., respectively*

$$\bar{m}(\mathcal{S}, \rho, (\overrightarrow{c}^P, \overrightarrow{c}^B)) := \frac{\sum_i \lambda_i \int_0^{c_i^P} m(\mathcal{S}, x, \rho, \sigma^*) f_i(x) dx}{\sum_i \lambda_i \int_0^{c_i^P} f_i(x) dx} \tag{100}$$

and

$$\bar{m}(\mathcal{S}, \rho, (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)) := \frac{\sum_i \lambda_i \left[\int_0^{c_i^L} m(\mathcal{S}, x, \rho, \sigma^*) f_i(x) dx + \int_{c_i^U}^{c_i^H} m(\mathcal{S}, x, \rho, \sigma^*) f_i(x) dx \right]}{\sum_i \lambda_i \left[\int_0^{c_i^L} f_i(x) dx + \int_{c_i^U}^{c_i^H} f_i(x) dx \right]}. \tag{101}$$

For every setting, $p_F < \mu v_1$ and $\varepsilon > 0$, there exists a (possibly fractional) number of spot instances $l_S \leq l$ such that for $\rho = (p_F, l_S)$ it holds that $\bar{m}(\mathcal{S}, \rho, \sigma^*)$ is greater than the expected payment in the fixed-price market minus ε, i.e.

$$\bar{m}(\mathcal{S}, \rho, \sigma^*) \geq \frac{p_F}{\mu} - \varepsilon. \tag{102}$$

Proof. For any fixed p_F there exists some number of spot instances l' such that all provider strategies $\rho = (p_F, l_S)$ with $0 < l_S \leq l'$ result in an equilibrium that is either of the form $\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$ or of the form $\sigma^* = (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$. We first present the case where $T \leq \frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right)$ for all $0 < l_S \leq l'$ and therefore $\sigma^* = (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$. To keep the proof readable, we introduce new notation to draw all waiting costs in the spot market from a single distribution instead of first drawing a job's class and then its waiting cost. Note that this does not change the number of jobs or their bids nor their waiting costs in the market. For provider strategy ρ and equilibrium strategy profile $\sigma^* = (\overrightarrow{c}^L, \overrightarrow{c}^U, \overrightarrow{c}^H)$, we define the distribution

$$F_S(c) \quad := \quad \frac{\sum_i \lambda_i \left[\int_0^{\min\{c, c_i^L\}} f_i(x) dx + \int_{\min\{c, c_i^U\}}^{c} f_i(x) dx \right]}{\sum_i \lambda_i \left[\int_0^{c_i^L} f_i(x) dx + \int_{c_i^U}^{c_i^H} f_i(x) dx \right]} \tag{103}$$

and the arrival rate

$$\lambda_S \quad := \quad \sum_i \lambda_i \left[\int_0^{c_i^L} f_i(x) dx + \int_{c_i^U}^{c_i^H} f_i(x) dx \right] \tag{104}$$

with similarly constructed PDF $f_S(c)$. Now consider an artificial spot market with arrival rate λ_S, where every arriving job's waiting cost is drawn from F_S and everyone joins. From the provider's point of view, this market is the same as the normal spot market that would result from her playing ρ, including users on average having the same expected payments. To analyze the provider's profit from the spot market when playing ρ, we can thus instead analyze this artificial market.

The per-user-average profit $\bar{m}(\mathcal{S}, \rho, \sigma^*)$ of the artificial spot market is given by taking the expectation of the payment $m(\mathcal{S}, c, \rho, \sigma^*)$, where the expectation is taken over c drawn from the

PDF $f_S(c)$:

$$\bar{m}(\mathcal{S},\rho,\sigma^*) = \frac{\lambda_S\left[\int_0^{c_1^H} m(\mathcal{S},x,\rho,\sigma^*)f_S(x)dx\right]}{\lambda_S} \tag{105}$$

$$= \int_0^{c_1^H} m(\mathcal{S},x,\rho,\sigma^*)f_S(x)dx \tag{106}$$

$$= \int_{-\infty}^{\infty} m(\mathcal{S},x,\rho,\sigma^*)f_S(x)dx \tag{107}$$

$$= E_{c\sim f_S}[m(\mathcal{S},c,\rho,\sigma^*)] \tag{108}$$

Now, for any l_S and any $0<\xi<1$ define $c_\xi^{l_S}$ as the waiting cost with $F_S(c_\xi^{l_S})=\xi$. It then follows by Markov's inequality that

$$\bar{m}(\mathcal{S},\rho,\sigma^*) \geq (1-F_S(c_\xi^{l_S}))m(\mathcal{S},c_\xi^{l_S},\rho,\sigma^*) \tag{109}$$

$$= (1-\xi)m(\mathcal{S},c_\xi^{l_S},\rho,\sigma^*). \tag{110}$$

Further, because the total waiting time is monotone increasing, by the integral upper bound, the following holds:

$$\frac{p_F}{\mu} = \int_0^{c^L} w(\mathcal{S},x,\rho,\sigma^*)dx + c^L(\frac{1}{\mu}-T) \tag{111}$$

$$\leq c^L \; max_{c\in[0,c^L]}\; w(\mathcal{S},c,\rho,\sigma^*) + c^L(\frac{1}{\mu}-T) \tag{112}$$

$$= c^L w(\mathcal{S},0,\rho,\sigma^*) + c^L(\frac{1}{\mu}-T) \tag{113}$$

Now observe that the cutoff point c^L goes to zero as the spot market becomes sufficiently small (i.e., $c^L \xrightarrow[l_S\to 0]{} 0$). Combined with Equation (113), it follows that the waiting time goes to infinity for users with bid 0, i.e.:

$$w(\mathcal{S},0,\rho,\sigma^*) \xrightarrow[l_S\to 0]{} \infty. \tag{114}$$

As a job with bid 0 is served exactly when there is an idle instance in the spot queue (i.e., there are fewer than l_S jobs of higher priority in the spot queue), the instance utilization of the spot queue has to go to full as the size of the spot market becomes sufficiently small, i.e.

$$\frac{\lambda_S}{l_S\mu} \xrightarrow[l_S\to 0]{} 1. \tag{115}$$

Now fix some $\xi>0$. It holds that

$$\frac{(1-F_S(c_\xi^{l_S}))\lambda_S}{l_S\mu} \xrightarrow[l_S\to 0]{} (1-\xi)1 \tag{116}$$

i.e., as the size of the spot market goes towards zero, the (average) utilization of the spot instances by jobs with priority over $c_\xi^{l_S}$ will always at most be $(1-\xi)$. For a given ξ (but independent of l_S), this limits the total waiting time at $c_\xi^{l_S}$ to some possibly very high but finite value $\bar{w}_\xi$. For any $c_\xi^{l_S}$ it

further either holds $c_\xi^{l_S} > c_1^L$ (and $m(\mathcal{S}, c_1^L, \rho, \sigma^*) < m(\mathcal{S}, c_\xi^{l_S}, \rho, \sigma^*)$ trivially) or the following holds:

$$m(\mathcal{S}, c_1^L, \rho, \sigma^*) \tag{117}$$

$$= \int_0^{c_\xi^{l_S}} w(\mathcal{S}, x, \rho, \sigma^*)dx + \int_{c_\xi^{l_S}}^{c_1^L} w(\mathcal{S}, x, \rho, \sigma^*)dx - c_1^L w(\mathcal{S}, c_1^L, \rho, \sigma^*) \tag{118}$$

$$\leq \int_0^{c_\xi^{l_S}} w(\mathcal{S}, x, \rho, \sigma^*)dx + (c_1^L - c_\xi^{l_S}) w(\mathcal{S}, c_\xi^{l_S}, \rho, \sigma^*) - c_1^L w(\mathcal{S}, c_1^L, \rho, \sigma^*) \tag{119}$$

$$= \int_0^{c_\xi^{l_S}} w(\mathcal{S}, x, \rho, \sigma^*)dx - c_\xi^{l_S} w(\mathcal{S}, c_\xi^{l_S}, \rho, \sigma^*) + c_1^L (w(\mathcal{S}, c_\xi^{l_S}, \rho, \sigma^*) - w(\mathcal{S}, c_1^L, \rho, \sigma^*)) \tag{120}$$

$$\leq \int_0^{c_\xi^{l_S}} w(\mathcal{S}, x, \rho, \sigma^*)dx - c_\xi^{l_S} w(\mathcal{S}, c_\xi^{l_S}, \rho, \sigma^*) + c_1^L (\bar{w}_\xi - w(\mathcal{S}, c_1^L, \rho, \sigma^*)) \tag{121}$$

$$= m(\mathcal{S}, c_\xi^{l_S}, \rho, \sigma^*) + c_1^L (\bar{w}_\xi - w(\mathcal{S}, c_1^L, \rho, \sigma^*)) \tag{122}$$

As $c_1^L \xrightarrow[l_S \to 0]{} 0$, it follows that, for all $0 < \xi < 1$ and all $\delta > 0$ there exists an l_S with $m(\mathcal{S}, c_\xi^{l_S}, \rho, \sigma^*) \geq m(\mathcal{S}, c_1^L, \rho, \sigma^*) - \delta$ and therefore

$$\bar{m}(\mathcal{S}, \rho, \sigma^*) \geq (1-\xi) m(\mathcal{S}, c_\xi^{l_S}, \rho, \sigma^*) \tag{123}$$

$$\geq (1-\xi)(m(\mathcal{S}, c_1^L, \rho, \sigma^*) - \delta) \tag{124}$$

Choosing ξ and δ such that $\frac{1}{2}\varepsilon \geq \xi m(\mathcal{S}, c_1^L, \rho, \sigma^*) + (1-\xi)\delta$ and noting that by Lemma A.2 it holds $m(\mathcal{S}, c_1^L, \rho, \sigma^*) = \frac{p_F}{\mu}$ then yields

$$\bar{m}(\mathcal{S}, \rho, \sigma^*) \geq m(\mathcal{S}, c_1^L, \rho, \sigma^*) - \frac{1}{2}\varepsilon = \frac{p_F}{\mu} - \frac{1}{2}\varepsilon \tag{125}$$

and the statement of the lemma follows.

When $T > \frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right)$ and $\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$, we analogously (only replacing the relevant notation) obtain

$$\bar{m}(\mathcal{S}, \rho, \sigma^*) \geq m(\mathcal{S}, c_1^P, \rho, \sigma^*) - \frac{1}{2}\varepsilon. \tag{126}$$

Because users with waiting cost $\overrightarrow{c}_1^P$ are indifferent between both markets, it has to hold that $\frac{p_F}{\mu} + c_1^P(\frac{1}{\mu} + T) = c_1^P \frac{1}{\mu}\frac{1}{1-\tau\psi_E(l_S)} + m(\mathcal{S}, c_1^P, \rho, \sigma^*)$. Solving this for $m(\mathcal{S}, c_1^P, \rho, \sigma^*)$ and substituting it into Equation (126) yields

$$\bar{m}(\mathcal{S}, \rho, \sigma^*) \geq \frac{p_F}{\mu} - c_1^P\left(\frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right) - T\right) - \frac{1}{2}\varepsilon. \tag{127}$$

Lastly note that $c_1^P\left(\frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right) - T\right) \xrightarrow[l_S \to 0]{} 0$ because $\frac{1}{\mu}\left(\frac{1}{1-\tau\psi_E(l_S)} - 1\right) - T$ increases in l_S and it holds that $c_1^P \xrightarrow[l_S \to 0]{} 0$. For any $\varepsilon > 0$, l_S small enough therefore yields the statement of the lemma. □

B BASIS OF THE NUMERICAL EXAMPLE

In this section, we give a precise description of how we calculate waiting times and preemption probabilities for the numerical examples. In the following we let $\varphi(l, \frac{\lambda}{\mu})$ denote the probability that

fewer than l jobs are currently in a queue with l instances, arrival rate λ and with expected job service time $\frac{1}{\mu}$. For memoryless queues, $\varphi(l, \frac{\lambda}{\mu})$ is given by the well-known Erlang C formula:[20]

$$\varphi(l, \frac{\lambda}{\mu}) = 1 - \left(1 + (1 - \frac{\lambda(c)}{l\mu}) \frac{l!}{\frac{\lambda(c)}{\mu}^l} \sum_{k=0}^{l-1} \frac{\frac{\lambda(c)}{\mu}^k}{k!} \right)^{-1} \tag{128}$$

Given this, the required calculations of the numerical example for the fixed-price queue are straightforward, while we need to make additional simplifying assumptions for the spot queue.

B.1 Fixed-price Queue

For a fixed-price queue, the total waiting time $w(\mathcal{F}, c, \rho, \sigma)$ and payment $m(\mathcal{F}, c, \rho, \sigma)$ are directly determined by the parameters of the setting. The only thing left to calculate is the minimal number of instances $l_F(\rho, \sigma)$ required to serve all users in the fixed-price market while observing the upper bound on the queuing time T. This can easily be done using the Erlang C formula, as it is well known that the expected queuing time $q(l, \frac{\lambda}{\mu})$ of a user joining a FIFO queue with l instances, arrival rate λ and service time $\frac{1}{\mu}$ is given by

$$q(l, \frac{\lambda}{\mu}) = \frac{1 - \varphi(l, \frac{\lambda}{\mu})}{l\mu - \lambda}. \tag{129}$$

See (Cooper 1981) for a proof. Plugging in the arrival rate into the fixed-price market for any pair of strategies (ρ, σ) and solving $q(l_F(\rho, \sigma)) = T$ then yields $l_F(\rho, \sigma)$.

B.2 Spot Queue

For the spot queue, the total waiting time $w(\mathcal{S}, c, \rho, \sigma)$ and payment $m(\mathcal{S}, c, \rho, \sigma)$ are not directly determined by the setting because they depend on the dynamics of the queue. Unfortunately, we cannot directly use the Erlang C to derive those terms because this would require that the running times and queuing times for all users are equal (Buzen and Bondi 1983). Since with priorities, and especially when costly preemptions are present, they are *not* equal, we make the following two simplifying assumptions (for the numerical examples only). This then allows us to calculate waiting times and preemption probabilities.

Assumption. *For calculating $w(\mathcal{S}, c, \rho, \sigma)$, we assume that jobs, while running, see the steady state distribution over states of the spot queue (when looking at the queue not including itself).*

Note that Assumption B.2 would be exactly satisfied if a given job ran for an infinite amount of time. Since jobs start in a random state but end in a state in which the queue has free capacity for a job of the same priority, jobs in practice see more "busy" states than in steady state and consequently take slightly longer to run. Importantly, we only make this assumption when calculating any single jobs's runtime, but we still calculate the steady state of the queue exactly to avoid the accumulation of approximation errors.

Assumption. *Any additional running time above and beyond a job's service time $\frac{1}{\mu}$ is run on "abstract" additional instances and does not influence the spot queue's steady state. However, while run on these abstract instances, a job still causes load-dependent costs for the provider and is still (internally and externally) preempted as if it was in the queue, as denoted by $\psi_I(c, \rho, \sigma)$ and $\psi_E(l_S)$.*

[20] See for example (Cooper 1981); a proof of the Erlang C formula can be found in (Takagi 2008).

Effectively, Assumption B.2 dynamically gives the spot market more instances than it actually has, to accommodate the additional running time needed due to preemptions.

Taken together, these two assumptions give us the following very useful Lemma.

Lemma B.1. *During its time in the spot queue, a job with bid c sees the steady state distribution of a FIFO queue with arrival rate $\lambda(c)$ and service time $\frac{1}{\mu}$. Here, $\lambda(c)$ denotes the arrival rate of jobs with a higher priority; i.e., during any time unit, on average, $\lambda(c)$ jobs with a higher priority than c arrive into the queue.*

Proof. Note that jobs with a lower bid do not influence the total waiting time of a user and can thus be ignored. As the probability that any other user in the system also has a waiting cost of exactly c is zero, we can assume that every other job has a strictly higher bid and thus a strictly higher priority. Combining this with Assumption B.2, we can assume that the job, while running, sees the steady state probabilities of the queue consisting of only those users with higher priorities than itself. Furthermore, by Assumption B.2, these steady state probabilities are the same as the steady state probabilities with zero preemption costs, which in turn are the same as the steady state probabilities of a FIFO queue consisting of all users with higher priority (see Buzen and Bondi (1983)). Taken together, the statement follows. □

Given Lemma B.1, we can now derive expressions for the waiting time and the expected number of internal preemptions per time unit.

Proposition B.2. *Given Assumptions B.2 and B.2, provider strategy $\rho = (p_F, l_S)$, and user strategy profile σ, the total waiting time of a user with bid c is given by*

$$w(\mathcal{S}, c, \rho, \sigma) \quad = \quad \frac{r(\mathcal{S}, c, \rho, \sigma)}{\varphi(l_S, \frac{\lambda(c)}{\mu})}, \tag{130}$$

where $\lambda(c)$ denotes the arrival rate of jobs with a bid higher than c into the spot queue (given σ).

Proof. Recall that the waiting time is defined as

$$w(\mathcal{S}, c, \rho, \sigma) \quad = \quad q(\mathcal{S}, c, \rho, \sigma) + r(\mathcal{S}, c, \rho, \sigma). \tag{131}$$

Observe that when a job with bid c is in the spot queue, it is run whenever there are fewer than l_S jobs with a higher bid in the system. By Lemma B.1, during its runtime, a jobs sees the steady state distribution of a FIFO queue with arrival rate $\lambda(c)$ and service time $\frac{1}{\mu}$. Thus, to be running for one full time unit, the job with bid c will, on average, have a waiting time of $\frac{1}{\varphi(l_S, \frac{\lambda(c)}{\mu})}$ time units. The statement of the Proposition now follows by noting that the running time of a job is given by $r(\mathcal{S}, c, \rho, \sigma)$. □

Proposition B.3. *Given Assumptions B.2 and B.2, provider strategy $\rho = (p_F, l_S)$, and user strategy profile σ, the expected number of internal preemptions per time unit of a user with bid c is given by*

$$\psi_I(c, \rho, \sigma) \quad = \quad \frac{(\mu l_S - \lambda(c))(1 - \varphi(l_S, \frac{\lambda(c)}{\mu}))}{\varphi(l_S, \frac{\lambda(c)}{\mu})}, \tag{132}$$

where $\lambda(c)$ denotes the arrival rate of jobs with a bid higher than c into the spot queue (given σ).

Proof. While a job with bid c is running, it will be internally preempted whenever the system contains exactly $l_S - 1$ jobs of higher priority and another job of higher priority arrives. By Lemma B.1, during its runtime, the job sees the steady state distribution of a FIFO queue with arrival rate $\lambda(c)$ and service time $\frac{1}{\mu}$. Thus, given a newly-arriving job (with priority higher than c), the

probability that this job preempts the job with bid c is equal to the probability that the FIFO queue contains exactly $l_S - 1$ jobs, which is given by

$$\frac{(1 - \frac{\lambda(c)}{\mu l_S})}{\frac{\lambda(c)}{\mu l_S}}(1 - \varphi(l_S, \frac{\lambda(c)}{\mu})) \tag{133}$$

(see (Cooper 1981)). Since we are interested in the preemption rate taken over the running time of the job (as opposed to the total time the job is in the system), we normalize this term by the probability that less than l_S jobs of higher priority are in the system. Because $\lambda(c)$ jobs with higher priority arrive per time unit, we also multiply with $\lambda(c)$, which yields

$$\psi_I(c, \rho, \sigma) = \lambda(c)\frac{\frac{(1-\frac{\lambda(c)}{\mu l_S})}{\frac{\lambda(c)}{\mu l_S}}(1 - \varphi(l_S, \frac{\lambda(c)}{\mu}))}{\varphi(l_S, \frac{\lambda(c)}{\mu})} \tag{134}$$

$$= \frac{(\mu l_S - \lambda(c))(1 - \varphi(l_S, \frac{\lambda(c)}{\mu}))}{\varphi(l_S, \frac{\lambda(c)}{\mu})}. \tag{135}$$

□

Taking the expression for the waiting time from Proposition B.2, plugging in the expression for the running time from Proposition 3.1, and lastly plugging in the expression for the internal preemptions from Proposition B.3, we can now write the expected total waiting time $w(\mathcal{S}, c, \rho, \sigma)$ of a user joining the spot queue as

$$w(\mathcal{S}, c, \rho, \sigma) = \frac{1}{\mu\varphi(l_S, \frac{\lambda(c)}{\mu})}\frac{1}{1 - \tau(\psi_I(c, \rho, \sigma) + \psi_E(l_S))} \tag{136}$$

$$= \frac{1}{\mu\varphi(l_S, \frac{\lambda(c)}{\mu})}\frac{1}{1 - \tau(\frac{(\mu l_S - \lambda(c))(1-\varphi(l_S, \frac{\lambda(c)}{\mu}))}{\varphi(l_S, \frac{\lambda(c)}{\mu})} + \psi_E(l_S))}. \tag{137}$$

Using the iterative approach described in the proof of Proposition 4.4, we can calculate the cutoff vectors of the user equilibrium strategies by solving a number of non-linear root searches. This allows us to calculate payments and profits and search for the optimal provider strategy $\rho = (p_F, l_S)$.

C USER WELFARE IN EXAMPLE 1

To help us better understand how the hybrid strategy with $p_F = p_F^{*0}$ affects the users, Figure 5 shows the user welfare for this strategy and compares it against the fixed-price strategy (we omit the other two strategies because plotting all four strategies makes the figure very hard to read). As we can see, for the fixed-price strategy (dashed blue line), the user welfare monotonically decreases in the instance costs κ. While the instance costs do not

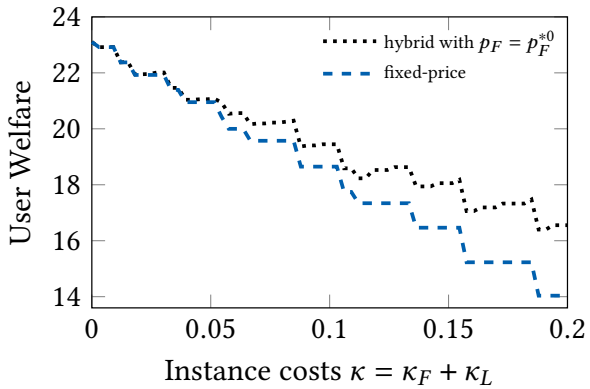

Fig. 5 User welfare under different strategies while varying the instance costs κ

directly affect the users, they influence the optimal provider strategies. This leads to the observed discontinuities in the welfare, since instances can only be bought in discrete units and the provider consequently changes her strategy in discrete steps.

It is clearly visible that the hybrid strategy with $p_F = p_F^{*0}$ (the dotted black line) shares many large discontinuity points with the fixed-price strategy (the dashed blue line). This happens because both strategies share the same price p_F^{*0}, which at these points increases, causing some users to balk and making everyone that remains in the fixed-price market worse off. However, the *size* of the discontinuities may differ, because under the hybrid strategy, some users might additionally move from the fixed-price to the spot market. Perhaps surprisingly, *between* these shared discontinuity points, the welfare corresponding to the hybrid strategy often increases. This happens because the provider is restricted to $p_F = p_F^{*0}$ and cannot optimally increase her price, so has to change l_S instead. To understand this in more detail, consider a situation where a cost increase does not lead to a change of p_F, but where the provider still wants users to move out of the relatively less profitable fixed-price market and into the spot market. In this situation, she then has to increase the attractiveness of the spot market, which increases the user welfare. Finally, as expected, the hybrid strategy with $p_F = p_F^{*0}$ always leads to a higher user welfare than the fixed-price strategy, which can be seen by the dotted black line always lying above the dashed blue line.

D USING IDLE FIXED-PRICE INSTANCES FOR THE SPOT MARKET

In this section, we consider an alternative model where the provider takes the spot instances from the pool of currently idle *fixed-price instances* instead of using other idle capacity (long-term reserved instances, maintenance capacity, etc.). In Section D.1., we first discuss the importance of using instances that are *reliably idle* for the spot market and why the fixed-price market is typically *not* the best source for such instances. Nevertheless, as some providers may want to use idle instances from the fixed-price market, we then show how the new cross-channel interactions change our results compared to our main model. In Section D.2, we first present the required changes to the model. In Section D.3, we then show how the equilibria of the user game change. Finally, in Section D.4, we then derive a similar condition for how a provider can simultaneously achieve a profit increase and a Pareto improvement for the users (as we did for our main model).

D.1 The Availability of Reliably Idle Instances from the Fixed-Price Market

As preemptions are costly for the users, the "usefulness" of an idle instance critically relies on how *reliably idle* it is (i.e., for how long the instance will remain idle). Therefore, the provider should use the most reliably idle instances for the spot market. While fixed-price markets of large cloud providers do contain a reasonable number of idle instances *on average*, only few of those instances are reliably idle and providers should not simply put any idle fixed-price instance on the spot market. This is due to two effects: larger markets require relatively smaller buffers and these buffers will be used more frequently the larger the market (but for increasingly shorter durations). To see this, we now provide a simple but striking numerical example.

Example D.1. Consider two M/M/l queues, one with arrival rate 100 and one with arrival rate 1000. Assume for both an expected service time of $\frac{1}{\mu} = 1$ and an SLA of $T = 0.0001$. We can use the Erlang C formula to derive that we need a buffer of 22 instances to satisfy the SLA for the first queue with arrival rate of 100 (i.e., the provider needs $l_F = 122$ fixed-price instances). For the second queue with arrival rate 1000 the required buffer only grows from 22 to 55 (i.e., the provider now needs $l_F = 1055$ fixed-price instances).

Now assume that the provider uses up to 1 idle instance for a spot queue, i.e., $l_s = 1$. Assume that this idle instance comes from the first queue (with arrival rate 100) and whenever the provider

has at least 1 idle fixed-price instance and no spot job is running, he starts a new spot job. Then a job in the spot market would in expectation get externally preempted at a rate of $\psi_E = 0.47$. For the second queue (with arrival rate 1000), the rate of external preemptions for spot jobs rises to $\psi_E = 3.07$. For both queues, the high preemption rate occurs because a large queue can frequently reach zero idle capacity while still satisfying the SLA, as it is highly likely that another instance becomes free shortly thereafter (and because newly-arriving users are willing to wait a little bit). However, any time this happens, the spot instance is immediately preempted. This effect becomes more pronounced the larger the fixed-price queue, which explains the large rise of the preemption rate for the second queue with arrival rate 1000.

In practice, the provider wants to keep the preemption rate reasonably low. To this end, she can decide to only start spot jobs whenever the expected time until any running spot job will be externally preempted is above some threshold t_E, resulting in $\psi_E < \frac{1}{t_E}$. While doing this reduces the number of external preemptions, this of course also further reduces the supply of instances for the spot market. For example, if the provider wanted to ensure that jobs (in expectation) run for at least $t_E = 20$ before they get preempted (leading to $\psi_E < \frac{1}{t_E} = 0.05$), then she would have to only start spot jobs when the fixed-price queue contains less than 105 jobs (for the queue with arrival rate 100 and $l_F = 122$ fixed-price instances) and less than 957 jobs (for the queue with arrival rate 1000 and $l_F = 1055$ fixed-price instances).[21]

Note that the size of the effects observed in the example are particularly large because the example considers memoryless service processes. While real-world service processes are usually heavy-tailed (which leads to larger and more reliably idle buffers), the observation that larger fixed-price markets lead to less reliably idle instances remains true in practice. Thus, a provider who wants to limit the number of external preemptions can only offer relatively few reliably idle fixed-price instances on the spot market. Additionally, the provider has to consider the cross-channel effects that occur when users move from the fixed-price market to the spot market, which decreases the number of fixed-price instances but not the number of users. While a provider can nevertheless choose to offer idle fixed-price instances on the spot market, most providers typically have access to many alternative instances that are more reliably idle than most idle instances from the fixed-price market. This includes instances from other business areas (e.g., long-term reserved instances), maintenance instances (which make up 5-10% of the capacity of a cloud computing center), etc. At least some of these alternatives are available for all current major cloud providers.

D.2 Required Model Changes

Even though most idle instances from the fixed-price market are typically not reliably idle, some cloud providers may still want to use them for a secondary spot market. Therefore, we now show how to adapt our model to using idle fixed-price instances. The most immediate change is that an *external preemption* now happens whenever a spot user is preempted in favor of a fixed-price user. Thus, while previously the number of external preemptions $\psi_E(l_S)$ was a function given by the setting that only depended on the provider's strategy ρ, the number of external preemptions $\psi_E(c, \rho, \sigma)$ now arises from the queueing system. Specifically, it now also depends on the strategies of all other users σ and, because lower bids get preempted first, also on a user's bid c. While in our main model, l_S denotes the (average) number of offered spot instances, it now denotes the maximum number of idle fixed-price instances the provider offers on the spot market, i.e., l_S is now an *upper bound* on the number of offered spot instances.

[21] The preemption rates and the expected time until the next preemption can be calculated by solving the difference equations of the corresponding Markov chains.

To control the number of external preemptions and only offer sufficiently reliably idle instances on the spot market, we introduce an additional strategy variable for the provider t_E, which denotes that the provider only starts a new spot job whenever, after starting this job, the expected time until the next external preemption for any running spot job is above the threshold t_E. Thus, given provider strategy $\rho = (p_F, l_S, t_E)$, for a job to be started in the spot market, four conditions have to be satisfied: (1) no job with a higher priority is waiting; (2) if the job started, there would be at most l_S spot jobs running, (3) there has to be an idle fixed-price instance or there is currently a spot job with a lower priority running, and (4) if the job started now, the expected time until the next external preemption for any running spot job would be at least t_E. Note that this implies that a spot job with low priority is *not* immediately preempted when a spot job with higher priority is waiting if the expected time until the next external preemption is currently too low. Due to the new cross-channel interactions that arise because the spot instances are now taken from the fixed-price market, both the running time $r(\mathcal{S}, c, \rho, \sigma)$ and the queuing time $q(\mathcal{S}, c, \rho, \sigma)$ in the spot market are now highly dependent on the number of users that join the fixed-price market. Additionally, if the threshold t_E is set too high (for a given user strategy profile σ), then the provider may never start a spot job (effectively not offering a spot market). A setting in our alternative model is now fully defined by $(n, v, \lambda, \mu, F, \tau, \kappa_F, \kappa_L, T)$ because the alternative model does not contain a maximum number of available spot instances l nor an exogenous function for the number of external preemptions.

Because larger fixed-price markets can have less reliably idle capacity than smaller markets (see Example D.1), we may observe the counterintuitive effect that the waiting time of the users with the highest priority in the spot market can decrease in the number of people that join the spot market. However, the overall costs of any user joining the spot market, i.e, $\int_0^c w(\mathcal{S}, x, \rho, \sigma)dx$ typically increase for any fixed c if more users move to the spot market. To see this, note that when users move from the fixed-price market to the spot market, the total number of instances decreases, but the number of users does not. Yet we cannot say with certainty that $\int_0^c w(\mathcal{S}, x, \rho, \sigma)dx$ *always* increases because the service discipline of the whole market is not work-conserving (i.e., there can be an idle instance even though jobs are waiting when the time until the next external preemption is too low) and these dynamics change whenever users move from the fixed-price to the spot market. While this effect is typically negligible compared to the reduction in the number of instances in the system, we cannot fully exclude the possibility that there could be some parameterizations for which there is a σ where $\int_0^c w(\mathcal{S}, x, \rho, \sigma)dx$ is *locally* decreasing in the number of users that choose the spot market. To avoid having to handle those cases (which do not change the form of the potential equilibria, but could in rare cases potentially lead to the existence of multiple equilibria) we therefore make the following assumption for the rest of the paper:

Assumption. *The overall cost of a user with any fixed bid c that joins the spot market, i.e. $\int_0^c w(\mathcal{S}, x, \rho, \sigma)dx$, increases if additional users (compared to σ) move to the spot market.*

D.3 Equilibria

Whenever some instances are actually offered on the spot market, we obtain an equilibrium structure similar to the one derived in Subsection 4.3.1:

Proposition D.2. *For any provider strategy $\rho = (p_F, l_S, t_E)$, in any BNE of the user game where any user joins the spot market, any equilibrium strategy profile is of the form $\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$. Here, $\sigma = (\overrightarrow{c}^P, \overrightarrow{c}^B)$ denotes that a user of class i with waiting cost c joins the spot market when $c < c_i^P \leq c_i^B$ and the fixed-price market when $c_i^P < c < c_i^B$; when $c > c_i^B$, he balks and does not join any market. The*

cutoff point c_1^P and the cutoff vector $\overrightarrow{c}^B$ are the unique solution to the following system of equations:

$$0 = c_1^P(T + \frac{1}{\mu}) + \frac{p_F}{\mu} - \int_0^{c_1^P} w(\mathcal{S}, x, \rho, (\overrightarrow{c}^P, \overrightarrow{c}^B))dx \tag{138}$$

$$0 = v_i - \min\left\{c_i^B(\frac{1}{\mu} + T) + \frac{p_F}{\mu}, \int_0^{c_i^B} w(\mathcal{S}, x, \rho, (\overrightarrow{c}^P, \overrightarrow{c}^B))dx\right\} \quad \forall i \in \{1, \dots, n\} \tag{139}$$

The rest of the cutoff vector $\overrightarrow{c}^P$ is given as $c_i^P = min(c_1^P, c_i^B)$.

Proof. Even users with the highest bid in the spot market have to queue longer than users in the fixed-price market, because (by definition) a user only gets served in the spot market when the fixed-price market has idle capacity. Thus, all users in the spot market are willing to pay strictly less than what they would have to pay in the fixed-price market. The remainder of the proof is equivalent to the proof of Proposition 4.3. □

While this gives us the structure of the equilibrium when some spot instances are offered and utilized, the following proposition tells us when that is the case.

Proposition D.3. *For any provider strategy $\rho = (p_F, l_S, t_E)$, the equilibrium strategy profile of the users is*

(1) *$\sigma^* = \overrightarrow{c}^F$ (i.e., no user joins the spot market, as described in Proposition 4.1) if and only if $l_S = 0$ or t_E is "too high," i.e., the fixed-price queue arising from $\sigma = \overrightarrow{c}^F$ has no state for which the expected time until the next external preemption of a hypothetically starting spot job would be more than t_E.*
(2) *$\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$ otherwise.*

Proof. Recall from Proposition 4.1 that $\sigma = \overrightarrow{c}^F$ is the equilibrium user strategy profile when no spot market is offered. We denote by $(\overrightarrow{x}, \overrightarrow{c}^F)$ a different strategy profile where any user of class i with waiting cost c joins the spot market if $c < x_i$. We now look at different provider strategies and classify the corresponding user equilibrium strategy profiles. If $l_S = 0$, then $\sigma^* = \overrightarrow{c}^F$ trivially. Now assume that the fixed-price queue arising from $\sigma = \overrightarrow{c}^F$ has no state for which the expected time until the next external preemption of a hypothetically starting spot job would be more than t_E. Then, as long as almost all users (i.e., all besides at most a null set) play $\sigma = \overrightarrow{c}^F$, the provider would never start a spot job, even if a single user deviated to the spot market and $l_S > 0$. Consequently, it holds that $\int_0^x w(\mathcal{S}, c, \rho, \sigma)dc = \infty$. By Assumption D.2, it immediately follows that $\int_0^x w(\mathcal{S}, c, \rho, (\overrightarrow{x}, \overrightarrow{c}^F))dc = \infty$ for any $\overrightarrow{x}$ and thus, in equilibrium, no user joins the spot market. On the other hand, if $l_s > 0$ and if the fixed-price queue arising from user strategy profile $\sigma = \overrightarrow{c}^F$ has a state for which the expected time until the next external preemption of a hypothetically starting spot job would be more than t_E, then users with waiting cost very close to 0 prefer the spot market and by Proposition D.2 it holds that $\sigma^* = (\overrightarrow{c}^P, \overrightarrow{c}^B)$. □

D.4 Well-behaved Settings: Increasing Provider Profit and User Welfare

We now show how the profit and welfare result from our main model translates to the alternative model. First note that the bound from Lemma 5.1 on the number of saved fixed-price instances per fixed-price user who moves to the spot market still holds, as the mechanics of the fixed-price market did not change. Next, we translate Lemma 5.2 to the alternative model.

Lemma D.4. *The average running time in the spot market (i.e., the left-hand side of the following inequality) is bounded above as follows:*

$$\frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} r(\mathcal{S},x,\rho,\sigma) f_i(x)dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x)dx} < (\frac{1}{\mu}+\tau)\frac{1}{1-\tau\frac{1}{t_E}} \tag{140}$$

Proof. Recall that $\psi_I(y,\rho,\sigma)r(\mathcal{S},y,\rho,\sigma)$ denotes the number of internal preemptions a job suffers in expectation. By the same arguments as in Lemma 5.2, it holds that

$$\frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} \psi_I(x,\rho,\sigma) r(\mathcal{S},x,\rho,\sigma) f_i(x)dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x)dx} < 1 \tag{141}$$

and

$$\begin{aligned} r_I(\mathcal{S},c,\rho,\sigma) &= \psi_I(c,\rho,\sigma) r(\mathcal{S},c,\rho,\sigma)\tau\frac{1}{1-\tau\psi_E(c,\rho,\sigma)} && (142)\\ &\leq \psi_I(c,\rho,\sigma) r(\mathcal{S},c,\rho,\sigma)\tau\frac{1}{1-\tau\frac{1}{t_E}}, && (143)\end{aligned}$$

where (143) follows from (142) because, whenever a job starts to run, the expected time until the next preemption is bounded by t_E. Combining these two inequalities we obtain

$$\frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} r_I(\mathcal{S},x,\rho,\sigma) f_i(x)dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} f_i(x)dx} < \frac{\tau}{1-\tau\frac{1}{t_E}}. \tag{144}$$

Similar as in the proof of Lemma 5.2, we finally obtain

$$\frac{\sum_i \lambda_i \int_{x:\sigma_{i,1}(x)=\mathcal{S}} r(\mathcal{S},x,\rho,\sigma) f_i(x)dx}{\sum_i \lambda_i \int_{x:\sigma_{i,1}(xx)=\mathcal{S}} f_i(x)dx} < (\frac{1}{\mu}+\tau)\frac{1}{1-\tau\frac{1}{t_E}}. \tag{145}$$

□

Given these bounds, we can now state a well-behavedness condition analogous to Definition 5.3 for our main model, where the new parameter t^w corresponds to a lower bound on the strategy variable t_E (capturing the reliablity of the fixed-price instances):

Definition D.5. We say that a setting $(n,v,\lambda,\mu,F,\tau,\kappa_F,\kappa_L,T)$ is *t^w-well-behaved* if t^w is the infimum over the t_E for which the following holds:

$$\frac{1+\tau\mu}{1-\tau\frac{1}{t_E}} - 1 < \frac{\kappa_F}{\kappa_L} \tag{146}$$

With this definition in hand, we can now show a profit and welfare result analogous to Theorem 5.4 for our main model.

Theorem D.6. *Given a t^w-well-behaved setting $(n,v,\lambda,\mu,F,\tau,\kappa_F,\kappa_L,T)$ and any fixed-price strategy $\rho_0 = (p_F^0, 0, \infty)$ that results in a positive profit and for which the queue arising from the corresponding equilibrium user strategy profile σ_0^* has any state for which the expected time until the next external preemption of a hypothetically starting spot job would be more than t^w, then there exists a strategy $\rho = (p_F^0, l_S, t_E)$ with the same price p_F^0, with $0 < l_S$ and with $t_E \geq t^w$ that yields a higher profit for the provider, i.e.,*

$$\Pi((p_F^0, l_S, t_E), \sigma^*) > \Pi((p_F^0, 0, \infty), \sigma_0^*), \tag{147}$$

and the same strategy also yields a Pareto improvement for the users, i.e.,

$$\forall i \in \{1,\dots,n\}\ \forall c \in [0,\mu v_i]: \quad \pi_i^c(\alpha,\beta,\rho,\sigma^*) \geq \pi_i^c(\alpha,\beta,\rho_0,\sigma_0^*),\ \textit{and} \tag{148}$$

$$\exists i \in \{1,\dots,n\}\ \exists c \in [0,\mu v_i]: \quad \pi_i^c(\alpha,\beta,\rho,\sigma^*) > \pi_i^c(\alpha,\beta,\rho_0,\sigma_0^*). \tag{149}$$

Proof. By Proposition D.3, any such strategy $\rho = (p_F^0, l_S, t_E)$ leads to some users joining the spot market in equilibrium. The proof of the theorem is then equivalent to the proof of Theorem 5.4 after replacing the general well-behavedness bound on the running time with $b(l_S) = \frac{1+\tau\mu}{1-\tau\frac{1}{t_E}}$. □

Informally, Theorem D.6 says that if the provider's current fixed-price market has some instances that are sufficiently reliably idle, then she can obtain a profit increase and achieve a Pareto improvement for the users by offering a spot market alongside her existing fixed-price market (as in our main model). Note that, in contrast to our main model, executing the provider's strategy in practice is now more difficult, because it will typically be intractable to exactly calculate, for every possible state, whether t_E would be satisfied when starting a new job. However, in this case, the provider could still approximate t_E (e.g., by using historical or simulated data).

While our analysis shows that offering idle fixed-price instances on the spot market can (in principle) be advantageous for the provider, recall from Section D.1 that a provider typically only has relatively few fixed-price instances that are sufficiently reliably idle. In contrast, instances from other areas of the cloud computing center (e.g., long-term reserved instances, maintenance instances, or capacity buffers intended for hardware failure) usually offer a better stock of idle capacity. We therefore recommend using idle instances from the fixed-price market only to bolster the supply of instances for the spot market when the utilization of the fixed-price market is particularly low and to instead primarily use other sources of idle capacity for the spot market.

3 On the cluster admission problem for cloud computing

The content of this chapter has previously appeared in:

> Ludwig Dierks, Ian Kash and Sven Seuken (accepted and forthcoming in 2021) **On the cluster admission problem for cloud computing**. *Journal of Artificial Intelligence Research*;
>
> Ludwig Dierks, Ian Kash and Sven Seuken (2019) **On the cluster admission problem for cloud computing**. *Proceedings of the 14th Workshop on the Economics of Networks.*

On the cluster admission problem for cloud computing*

Ludwig Dierks DIERKS@IFI.UZH.CH
University of Zurich

Ian A. Kash† IANKASH@UIC.EDU
University of Illinois at Chicago

Sven Seuken SEUKEN@IFI.UZH.CH
University of Zurich

Abstract

Cloud computing providers face the problem of matching heterogeneous customer workloads to resources that will serve them. This is particularly challenging if customers, who are already running a job on a cluster, scale their resource usage up and down over time. The provider therefore has to continuously decide whether she can add additional workloads to a given cluster or if doing so would impact existing workloads' ability to scale. Currently, this is often done using simple threshold policies to reserve large parts of each cluster, which leads to low efficiency (i.e., low average utilization of the cluster). We propose more sophisticated policies for controlling admission to a cluster and demonstrate that they significantly increase cluster utilization. We first introduce the cluster admission problem and formalize it as a constrained Partially Observable Markov Decision Process (POMDP). As it is infeasible to solve the POMDP optimally, we then systematically design admission policies that estimate moments of each workload's distribution of future resource usage. Via extensive simulations grounded in a trace from Microsoft Azure, we show that our admission policies lead to a substantial improvement over the simple threshold policy. We then show that substantial further gains are possible if high-quality information is available about arriving workloads. Based on this, we propose an information elicitation approach to incentivize users to provide this information and simulate its effects.

1. Introduction

Cloud computing is a fast expanding market with high competition where small efficiency gains translate to multi-billion dollar profits.[1] Like many other markets (e.g., ridesharing platforms, kidney exchanges, online labor markets, and display advertising), the efficiency of this market relies on the performance of a matching algorithm (Ashlagi et al., 2019; Ma & Simchi-Levi, 2019; Behnezhad & Reyhani, 2018; Assadi et al., 2017). In the cloud computing case, the matching algorithm matches incoming requests for virtual machines to the hardware that will be used to satisfy them.

Despite the importance of this matching, most cloud clusters currently run at low efficiency. In the cloud domain, low efficiency means low *average utilization* of the cluster (i.e., only a relatively small fraction of resources are actually used by customers at any given time). There are many reasons for this (Yan et al., 2016). These include technical limita-

∗. An early version of this work has appeared as a 6-page extended abstract in the proceedings of the 14th Workshop on the Economics of Networks, Systems and Computation (NetEcon'19).

†. A substantial portion of this work was done while this author was employed by Microsoft.

1. https://www.microsoft.com/en-us/Investor/earnings/FY-2018-Q2/press-release-webcast

tions (such as the need to reserve capacity for node failures or maintenance), inefficiencies in scheduling procedures (especially if virtual machines (VMs) might change size or do not use all of their requested capacity), as well as factors that are external to the cluster (such as fluctuations in overall demand). Another important cause is the nature of many modern workloads: highly connected tasks running on different VMs that should be run on one cluster to minimize latency and bandwidth use (Cortez et al., 2017). In practice, this means that different VMs from one user are bundled together into a *deployment* of interdependent workload. When the workload of a user changes, his deployment can request a *scale out* in the form of additional VMs or shut some of its active VMs down.

In this paper, we pay special attention to these size changes. Changing deployment sizes mean that providers face the difficult problem of deciding to which cluster to assign a deployment, as a deployment which is small today may, without warning, see a dramatic increase in size that must be accommodated. To get a sense of the difficulty of this problem, consider that, over time, the number of VMs needed by a specific deployment could vary by a factor 10 or even 100, and a request to scale out should almost always be accepted on the same cluster, as denying it would impair the quality of the service, possibly alienating customers. Furthermore, once a deployment is running in one cluster, it would be error prone to move it to a different cluster, making such migrations only feasible as a last resort (e.g., because of hardware failure). Providers consequently hold large parts of each cluster as idle reserves to guarantee that only a very low percentage of these scale out requests is ever denied, leading to relatively low average utilization.

1.1 Cluster Admission Control

We reduce the problem of determining to which cluster to assign a new deployment to the problem of determining, for a particular cluster, whether it is safe to *admit a deployment*, or if doing so would risk running out of capacity if some deployments scale (Cortez et al., 2017). While a lot of research has been done on scheduling *inside* the cluster (Schwarzkopf et al., 2013; Verma et al., 2015; Tumanov et al., 2016; Zhao et al., 2016), the admission problem has not been well studied before. Consequently, cloud providers are often still using simple policies like rejecting all new deployments once a cluster passes a fixed utilization threshold, effectively reserving a percentage of the cluster only for scale-outs.These may seem reasonable at first glance, as the law of large numbers might seem to suggest that with many jobs in a large cluster the current utilization would be a good guide to future utilization. But as Cortez et al. (2017) have shown, a relatively small number of deployments account for most of the utilization. This suggests that the *types of deployments* (i.e., small/large, fast/slow scaling, short/long lived etc.) currently in a cluster have a larger impact on the failure probability than is apparent, and policies that only take the current utilization into account are suboptimal.

1.2 Overview of Contributions

We formalize the cluster admission problem as a constrained Partially Observable Markov Decision Process (POMDP) (Smallwood & Sondik, 1973) where each deployment behaves according to some stochastic process and the cluster tries to maximize the number of active compute cores without exceeding its capacity. Since the exact stochastic processes of

individual arriving deployments are not known to the cluster, it has to reason about the observed behavior. The large scale of the problem as well as the highly complicated underlying stochastic processes make finding optimal policies infeasible, even for the underlying (fully observable) Markov Decision Process and with limited look-ahead horizon.

Since optimally solving this POMDP is not feasible, we next propose a strategy for constructing heuristic policies via a series of simplifying assumptions. These assumptions reduce the highly branching look-ahead space down to the approximation of a random variable using its moments. We then present the currently used threshold policy that does not take probabilistic information into account as well as two new policies that take successively higher moments into account. We fit our model to data from a real-world cloud computing center (Microsoft Azure internal jobs (Cortez et al., 2017)) and, via simulations, show that our higher moment policies produce a 30% improvement over current practice, which would translate to hundreds of millions of dollars a year in savings for large cloud providers.

In our basic model, relatively little is known about arriving deployments, so the performance gains we observe from our more sophisticated policies are driven by being able to better condition admission decisions on the current state of the cluster. We next examine how the utilization of the cluster can be further increased if more precise *prior information* about arriving deployments is available. Prior work has explored similar opportunities in the context of resource planning and scheduling in analytics clusters (Jyothi et al., 2016; Rajan et al., 2016). To study the value of prior information, we introduce a simple framework which captures a notion of the quality of information available. Through additional simulations, we quantify how our policies benefit from this additional information. Depending on the quality of information available, the resulting gains increase to $50\% - 65\%$ relative to current practice.

Finally, given the importance of the quality of this information, we design a new information elicitation mechanism, with the goal of simultaneously improving the cluster's utilization as well as the customers' utility. This requires care to find a design that allows meaningful information to be elicited in an incentive compatible way while being simple for customers to use. To this end, we propose that rather than explicitly asking users to describe the behavior of their deployments, cloud providers instead provide them with the opportunity to group their deployments into customer-defined categories with similar characteristics. The cloud provider can then set a small portion of the fee for a deployment using a pricing rule based on the variance of resource demands of deployments in a category. We show that such variance-based pricing provides users with the right incentives to (a) label their deployments properly (into, e.g., high and low variance deployments) and (b) structure their workloads in a way that helps the cluster run more efficiently. We provide additional simulations to quantify the benefits of an accurate labeling.

In practice, the magnitude of the gains from our approach will depend on many details our simulations elide. However, we believe that our simulations provide a persuasive case that (a) there are substantial economic gains available from using our new admission policies for the process of matching deployments to clusters and (b) there are substantial further improvements possible by using our information elicitation approach to elicit relevant information from customers.

While our work focuses on a specific problem faced by cloud providers, our overall approach is fundamentally about managing the tail risks of a stochastic process. In our case, these are the rare events where the cluster runs out of capacity. Thus, our approach may also be of interest in other domains where the management of tail risks is important, for example in finance.

1.3 Related work

There is a large literature on *cluster scheduling and load balancing* (Schwarzkopf et al., 2013; Verma et al., 2015; Tumanov et al., 2016; Wolke et al., 2015). In addition, some work addresses a different notion of admission control to a cluster, namely how to manage queues for workloads which will ultimately be deployed to that cluster (Delimitrou et al., 2013). In our work, however, while studying which deployments to admit to a cluster, we abstract away from the question of exactly which resources should be used, so our research is orthogonal to this prior work on scheduling and load balancing.

There is also a literature that views scheduling through the lens of stochastic online bin packing (Cohen et al., 2019; Song et al., 2014). This literature also deals with issues of changing workloads on possibly overcommitted resources. However, the models in these papers operate at smaller scales and shorter time horizons. At these scales, the key phenomena we study are not present.

One of the core characteristics of the cluster admission problem is that the arrival of a new deployment into the cluster causes an increase in scale-outs in the future (i.e., until the deployment dies). This effect of "work creating more work" is also broadly reminiscent of mutually-exciting (Hawkes) processes (see (Hawkes, 2018) for a recent survey). These have also been studied in the context of queuing systems (Daw & Pender, 2018), though there are some important differences (e.g., in our problem only accepted deployments cause scale outs and the rate of additional scale-outs varies greatly for different types of deployments).

There is a large literature on market design challenges in the context of the cloud (Kash & Key, 2016). Existing work has studied both queueing models where decisions are made online with no consideration of the future (Abhishek et al., 2012; Dierks & Seuken, 2019) and reservation models which assume very strong information about the future (Azar et al., 2015; Babaioff et al., 2017). Our work sits in an interesting intermediate position where users may have rough information about the types of their deployments. Furthermore, this literature focuses on using prices to determine which jobs should be served and which should not. While our problem is similarly about accepting or rejecting deployments, we do not want to ration through price discrimination. This is because (nearly) every request is ultimately served by the cloud computing provider and whether it goes into this or another cluster is of little consequence for the user.

Other market design work has looked at how multidimensional resources can be fairly divided among deployments. For example, *Dominant Resource Fairness* (Ghodsi et al., 2011) is an approach that has proven useful in practice (Hindman et al., 2011) and has inspired follow-up work more broadly in the literature on fair division (Parkes et al., 2015; Dolev et al., 2012; Gutman & Nisan, 2012; Kash et al., 2014). In our work, we assume that compute cores are the resource bottleneck and we do not model multi-dimensional resource

requirements. Therefore, the considerations studied in the above papers are not present in our work.

Solving POMDPs is a well-studied problem (Smith & Simmons, 2005; Russell & Norvig, 2016; Roy et al., 2005). Unfortunately, finding an optimal policy is known to be PSPACE-complete even for finite-horizon problems (Papadimitriou & Tsitsiklis, 1987). Even finding ϵ-optimal policies is NP-hard for any fixed ϵ (Lusena et al., 2001). In our case, the problem is further exacerbated by the existence of side constraints. Constrained POMDPS are far less well studied than unconstrained POMDPS. General (approximation) strategies proposed in the past include linear programming (Poupart et al., 2015; Walraven & Spaan, 2018), point-based value iteration (Kim et al., 2011), a mix of online-look ahead and offline risk evaluation (Undurti & How, 2010), and forward search with pruning (Santana et al., 2016). None of these approaches is efficiently applicable when the state space of the underlying MDP is large or, as in our case, partly continuous. Khonji, Jasour, and Williams (2019) recently proposed a fully polynomial time approximation scheme (FPTAS) for constant horizon constrained POMDPs. While their algorithm is polynomial in the size of the observation and action spaces, it is exponential in the number of time steps. This makes it not applicable in domains with long time horizons like cluster admission control. While some work has addressed continuous state space POMDPs (Porta et al., 2006; Duff & Barto, 2002; Brooks et al., 2006), none of this prior work is directly applicable to a constrained problem of the size we study in this paper.

2. Preliminaries

In this section, we formally model the cluster admission problem and then introduce a POMDP formulation to solve the provider's control problem.

2.1 Formal Model

We consider a single cluster in a cloud computing center. A cluster consists of c *cores* that are available to perform work, also called the cluster's *capacity*. These cores are used by *deployments*, i.e., interdependent workloads that use one or more cores. The set of deployments currently in the cluster is denoted by X, and each deployment $x \in X$ is assigned a number of cores C^x. Any core that is assigned to a deployment is called *active*, while the remainder are called *inactive*.[2] We do not model the exact placement of cores inside the cluster and in consequence we also do not model the grouping of cores into VMs.

A deployment can request to *scale out*, i.e., increase its number of active cores. Each such request is for one or more additional cores and must be accepted whenever activating the requested number of cores would not make the cluster run over capacity. Following current practice, scale out requests must be granted entirely or not at all. Deployments may shut down some of their cores over time and these cores then become inactive. A deployment *dies* when its number of active cores becomes zero. This can happen in two ways. First, it can die by successively shutting down one core after another until reaching

2. We assume that inactive cores can become active at any time. This means that features such as hardware failure or capacity reserved for maintenance are not modeled. This is a reasonable simplification, as they do not significantly affect the relative utilization of policies.

zero active cores. Second, it can die spontaneously by shutting down all of its cores at once; intuitively this models a decision by a user to kill the deployment.[3]

A deployment x is described by 4 deployment parameters $(C^x, \mu_x, \lambda_x, \sigma_x)$. We have already introduced the size of a deployment C^x. The remaining three parameters are drawn independently from population-wide distributions with PDFs $f_\lambda, f_\mu, f_\sigma$. We now explain how these parameters govern the behavior of the deployment.

At a high level, we assume that the deployments are memoryless (i.e., the basic processes governing a deployment's behavior only depend on the current state, which results in all processes following Poisson/exponential distributions). This is common in the literature whenever arrival and departure processes are modelled (e.g., in queuing theory), and has been used in previous models of cloud computing (Abhishek et al., 2012; Dierks & Seuken, 2019). Memorylessness is reasonable at cloud scale and simplifies some calculations, but it is not essential for our approach and policies.

Specifically, we assume that each core's *lifetime* is distributed according to an exponential distribution with parameter μ_x. The *maximum lifetime* of a deployment (i.e., the time between arrival and it spontaneously shutting down all of its remaining cores) is distributed according to an exponential distribution with parameter $\Delta\mu_x$, where Δ is a (population-wide) multiplicative factor. This effectively leads to an average maximum lifetime for the deployment of $\frac{1}{\Delta}$ average core lifetimes. The *number of scale outs* per time unit for the deployment x is distributed according to a Poisson distribution with rate parameter $\lambda_x \mu_x^\nu$, where ν is a population-wide parameter. This form of the rate parameter captures the empirical fact that deployments with longer-lived cores scale slower than those with short lived cores. The *size of a scale out* is distributed according to one plus a Poisson distribution with parameter σ_x. While this is an approximation on an individual level (VM sizes usually come in powers of 2), it is reasonable at the level of a cluster.

New deployment requests arrive over time and are accepted or rejected according to an *admission policy*.[4] The policy must limit the admission of new deployments to ensure that the cluster is not forced to reject a higher percentage of scale out requests than is specified by an internal *service level agreement* (SLA) τ.[5] If a scale out request cannot be accepted because the cluster is already at capacity, one failure for the purpose of meeting the SLA is logged. An optimal policy therefore maximizes the *utilization* of the cluster, i.e., the average number of active cores, while making sure the SLA is observed in expectation.

2.2 The Provider's Control Problem: POMDP Formulation

The problem the provider is facing when deciding whether to admit a deployment is that the decision must be made under uncertainty regarding future arrivals and the future behavior of deployments. In addition, the provider cannot directly observe the parameters of each deployment's processes. To understand how a provider can find a policy given

3. We model death as permanent because with no active cores any future request could be assigned to a different cluster.
4. A rejected deployment is only rejected from this cluster, not from the cloud computing center as a whole. While outside of our model, in practice it then simply gets sent to the next cluster.
5. This is a cluster-level SLA and not a deployment-level SLA, as in practice, the probability of tail-events such as scale out failures cannot feasibly be evaluated for a single deployment.

this uncertainty, we model the problem as a Partially Observable Markov Decision Process (POMDP)$(\mathbb{S}, \mathbb{A}, \mathbb{R}, \mathbb{T}, \Omega, \mathbb{O})$ whose policy is constrained to meet the SLA τ.

For the POMDP formulation, we assume that time is discrete[6] and that the problem has a finite time horizon[7] denoted N. The state space, denoted $\mathbb{S}$, describes the space of all possible states of the cluster. A state $s \in \mathbb{S}$ contains all information about the cluster's active deployments $X(s)$ (including, for each deployment x both its current size C^x and its scaling process parameters λ_x, μ_x and σ_x) as well as the deployments that arrived during the current time step. The action set $\mathbb{A}$ consists of individually accepting or rejecting each of the deployments that arrived this time step. The reward function $\mathbb{R}(s) = \sum_{x \in X(s)} C^x$ is the number of active cores in a state s. The transition probability function is denoted $\mathbb{T}(s'|s,a) \forall s's \in \mathbb{S}, \forall a \in \mathbb{A}$. Given a state of the cluster and admission decisions, this function captures the distribution over scale outs, core deaths, and arrivals of new deployments that occur during the next time step. Ω is the set of possible observations and $\mathbb{O} : \Omega \times \mathbb{S} \to [0,1]$ an observation model. In our case, the observation model $\mathbb{O}$ is deterministic, but many states share the same observation. For state s, we always observe $\omega \in \Omega$ equal to the sizes of all deployments that are in the cluster and those that arrived with the last state transition.

As is standard, we further denote the cluster's current knowledge about which state s it is in via a belief state $b \in \mathbb{B}$, i.e., a probability distribution over states. Specifically, a belief state b specifies, for each deployment x that is in the cluster or arrived with the last state transition, its current size C^x and (posterior) distributions $f^x_\lambda, f^x_\mu, f^x_\sigma$ over its scaling process parameters. For a given x, we let $\tilde{x} = (C^x, f^x_\lambda, f^x_\mu, f^x_\sigma)$, i.e., the provider's belief about the deployment x. A policy π can now be defined as a mapping from belief states to actions.

Whenever the cluster obtains a new observation $\omega \in \Omega$ in time step $n+1$, the belief state is updated according to the observation and transition models, i.e.,

$$b_{n+1}(s'|b_n, a, \omega) \propto \mathbb{O}(\omega|s') \sum_s \mathbb{T}(s'|s,a) b_n(s). \tag{1}$$

Given this, we can now define two auxiliary functions. We let $g_{n,\pi,b}$ denote the probability density function of the distribution over the states s_n and belief states b_n of the system n time steps in the future, given policy π and starting belief b. Furthermore, we let $h(s_n, \pi(b_n))$ denote the expected percentage of scale-outs that fail with the next state transition from a given state-action pair. We can now formulate the provider's control problem as finding an optimal policy given an SLA.

Problem 1 (Cluster Admission Problem)**.** *The cluster admission problem is to find an optimal policy π for the POMDP $(\mathbb{S}, \mathbb{A}, \mathbb{R}, \mathbb{T}, \Omega, \mathbb{O})$ subject to the following two constraints:*

$$\int_{(s_n, b_n)} g_{n,\pi,b}(s_n, b_n) h(s_n, \pi(b_n)) d(s_n, b_n) \leq \tau \quad \forall \textit{ safe } b \, \forall 0 \leq n < N \tag{2}$$

$$\pi(b) = \textit{reject all arrivals} \quad \forall \textit{ unsafe } b \tag{3}$$

6. While deployments can arrive at arbitrary times, it takes time to make the acceptance decision. Thus, there is little loss in discretizing time.
7. Our approach works for any choice of horizon (or even an infinite horizon with average or discounted rewards).

Here, we call belief state b safe *if the policy π_0 which always rejects newly arriving deployments satisfies*

$$\int_{(s_n,b_n)} g_{n,\pi_0,b}(s_n,b_n)h(s_n,\pi_0(b_n))d(s_n,b_n) \leq \tau \quad \forall 0 \leq n < N \tag{4}$$

and unsafe *otherwise.*

Intuitively, we would like our SLA constraint (2) to hold in every belief state. However, even if we follow an optimal policy, we can reach belief states where (in retrospect) too many deployments have been admitted, such that, even if no new deployments are admitted ever again, the constraint (2) would be violated. Thus, if we would require Equation (2) to hold in all belief states, we would have an infeasible problem. To address this, we do not enforce Equation (2) in *unsafe* belief states (as defined in Problem 1). We instead require the policy to reject all arriving deployments until it reaches a *safe* belief state.[8]

Note that the current time step is not referenced in Equation (2) or (4). This is intentional to avoid horizon effects: a cluster should not aggressively start to accept new deployments close to the end of its lifetime.

3. A Tractable Problem Formulation

Optimal policies for the cluster admission problem (i.e., Problem 1) cannot be calculated in practice for three reasons. First, there is no simple closed form for the state transition probabilities. Second, the state space of the the POMDP is very large: consider a cluster with 20,000 cores. It usually has hundreds of deployments, each described by 4 parameters, some of which are continuous. Even discretized, this results in a state space exponential in the size of the cluster. Third, even disregarding unlikely state transitions, the branching factor is large. This renders standard methods that rely on optimizing limited lookaheads infeasible. Therefore, we now present three carefully chosen simplifying assumptions under which we characterize an optimal policy. In Section 4, we use this characterization to design practical admission control policies.

Assumption 1 (No Future Arrivals)**.** *No further deployments arrive after the current timestep.*

This assumption ensures that a policy does not reject deployments simply because better behaved deployments might arrive in the future. In the cloud domain, this behavior is desirable, as even customers with high demand variability must be served by some cluster in the data center.

Assumption 2 (Relaxed Capacity Constraints)**.** *Deployments can scale out even if doing so exceeds the cluster capacity c. For the purpose of defining h, a scale out is considered to fail only if the cluster has already exceeded its capacity.*

With no future arrivals, the cluster's future state only depends on how the sizes of the currently active deployments change. However, if a cluster is full, further scale out requests

8. The requirement to reject *all* deployments is a design decision we revisit in Section 6.

by deployments are denied, introducing correlations between the future sizes of different deployments. Assumption 2 removes this correlation. In particular, let L_n^x denote the random variable that is the number of active cores of deployment x in time step n. With the first two assumptions, L_n^x is independent of $L_{n'}^{x'}$ for all $x \neq x'$ and all n, n'. The same holds for the random variable $L_n^{\tilde{x}}$ for the provider's belief. This is reasonable because the cluster being full should be rare if the SLA is being met.

Assumption 3 (At Most one Event per Timestep)**.** *In any timestep, at most one event occurs (i.e., at most one deployment scales out, shuts down cores, or arrives to the cluster).*

Since the probability that more than one event occurs in a single time step approaches zero with increased granularity of the time discretization, it is reasonable to assume this.

Using these three assumptions, we can now simplify the problem of determining when the SLA constraint is met. Recall that $\tilde{x} = (C^x, f_\lambda^x, f_\mu^x, f_\sigma^x)$ specifies the provider's belief over a deployment x. In the following, we denote by $A_\pi(b)$ the set of beliefs $\tilde{x}$ over the active deployments in belief state b and the deployments that are accepted with policy π in belief state b.

Proposition 1. *For all policies π, under Assumptions 1, 2 and 3, the following holds:*

$$\int_{(s_n,b_n)} g_{n,\pi,b}(s_n,b_n)h(s_n,\pi(b_n))d(s_n,b_n) = Pr(\sum_{\tilde{x}\in A_\pi(b)} L_n^{\tilde{x}} > c) \quad \forall b\, \forall 0 \leq n < N. \tag{5}$$

Proof. To see the that Equation (5) holds, note the following: By Assumption 1, it suffices to consider only the deployments that are currently in the cluster or arrive in the current time step, i.e., $\tilde{x} \in A_\pi(b)$. By Assumption 3, at most one scale out can fail per time step. Thus the left hand side of Equation (5) captures the following: if a scale out occurs in time step n, what is the probability that it fails. By Assumption 2, a scale out fails exactly when $\sum_{\tilde{x}\in A_\pi(b)} L_n^{\tilde{x}} > c$. □

Using this result, it is straightforward to characterize an optimal policy for the simplified problem.

Corollary 1. *Under Assumptions 1, 2 and 3, an optimal policy π accepts an arriving deployment in belief state b if and only if*

$$Pr(\sum_{\tilde{x}\in A_\pi(b)} L_n^{\tilde{x}} > c) \leq \tau \quad \forall 0 \leq n < N. \tag{6}$$

Proof. By Assumption 3, it suffices to consider one arrival. By Assumption 1, if an arrival could be accepted without violating the constraint, doing so is optimal. By Proposition 1, the constraint is equal to Inequality (6). □

Proposition 1 and Corollary 1 show that to implement an optimal policy for the simplified problem it suffices to evaluate the probability that a sum of independent random variables exceeds a threshold. The remaining question now is how to compute or approximate this probability for our complex processes fast enough to allow rapid responses to customer requests.

4. Designing new Admission Control Policies

In this section, we first define the complex random variables $L_n^{\tilde{x}}$ in terms of simpler random variables that directly arise from the processes. The essence of our approach is to take this definition of L_n and use it to compute approximate moments of L_n (i.e., approximate summary statistics of the behavior of the random variable). We then use these approximate moments to design new policies.

We can describe $L_n^{\tilde{x}}$ using the following random variables (which have a superscript $\tilde{x}$ which we generally omit for brevity):

- C is the variable denoting the number of active cores at time step 0.
- Y_i is the random variable denoting the number of scale outs that occur between time step $i-1$ and time step i, assuming the deployment has not died.
- $S_{i,l}$ is the size the l'th scale out request would have, assuming at least l scale out requests occur between time step $i-1$ and time step i.
- $Z_{n,i,k}$ is the binary random variable denoting whether the k'th core activated between time steps $i-1$ and i would still be active in time step n, assuming at least k cores were activated and the deployment has not died. For $i=0$, this instead refers to whether the k'th core that is active at timestep 0 is still active at time step n.
- D_i is the random variable which is 1 if x would not have died due to a lack of active cores before time step i. It can be defined recursively as

$$D_i = D_{i-1}(1-\Pi_{k=1}^{C}(1-Z_{i,0,k})\Pi_{j=1}^{i-1}\Pi_{k=1}^{\sum_{l=0}^{Y_i} S_{i,l}}(1-Z_{i,j,k})) \quad (7)$$

$$D_1 = 1-\Pi_{k=1}^{C}(1-Z_{1,0,k}) \quad (8)$$

- B_n is the random variable denoting the number of cores that were active at time step 0 and are still active in time step n, which can be calculated as

$$B_n = \sum_{k=1}^{C} Z_{n,0,k}. \quad (9)$$

- Q_n is the random variable denoting the number of cores activated between time step 0 and time step n that are still active assuming no service termination, i.e.

$$Q_n = \sum_{i=1}^{n}\sum_{k=1}^{\sum_{l=0}^{Y_i} S_{i,l}} Z_{n,i,k}. \quad (10)$$

- M_i is the random variable which is 1 if the maximum lifetime of the deployment is at least i and 0 otherwise.
- Finally, L_n can be calculated as $L_n = M_n D_n(Q_n + B_n)$.

We now turn to the design of our approximate policies.

4.1 Baseline (Zeroth Moment Policy)

Before introducing our new policies, we state the baseline admission control policy that is widely used in practice. It is a myopic policy that simply compares the current number of active cores to a threshold. This policy does not use any information about the set of deployments besides the total number of active cores. It can be we viewed as a degenerate case of our approach, as it does not take any probabilistic information about the random variables into account. We therefore also call it a *Zeroth Moment Policy*. Because it uses a limited amount of information, it must be conservative in how many deployments it accepts, since it does not know how often or fast they will scale out.

Definition 1 (Zeroth Moment Policy (Baseline))**.** *Under a zeroth moment policy π with threshold t, a newly arriving deployment is accepted if, after accepting the deployment, there would be less than t cores active.*

4.2 First Moment Policy

Our first policy approximates the probability of scale out failures (i.e., Equation (5)) by utilizing the first moments, i.e. the expected value of the deployment processes. By Markovs's Inequality, for a non-negative random variable L and $c \geq 0$, it holds that

$$Pr(L \geq c) \quad \leq \quad \frac{E[L]}{c}. \tag{11}$$

Such a policy that utilizes Markov's Inequality therefore rejects an arriving deployment when the expected utilization lies above a chosen threshold and otherwise accept.

Definition 2 (First Moment Policy)**.** *Under a first moment policy π with threshold t, a newly arriving deployment in belief state b is accepted if, after accepting the deployment, the expected number of active cores would be less than t in all future time steps, i.e.*

$$\sum_{\tilde{x} \in A_\pi(b)} E[L_n^{\tilde{x}}] \leq t \quad \forall 0 \leq n < N, \tag{12}$$

where $E[L_n^{\tilde{x}}]$ is approximated, for example using the approach described in Proposition 2.

Proposition 2. *Assuming all M_n, D_i, Q_n, and B_n are uncorrelated and $Z_{i,j,k}$, Y_i, and $S_{i,l}$ are uncorrelated as constituents of D_i, it holds:*

$$\begin{aligned}
E[L_n] &= E[M_n]E[D_n](E[Q_n] + E[B_n]) && (13)\\
E[Q_n] &= \sum_{i=1}^{n} E[E[Y_1|\lambda,\mu]E[S_{1,1}|\sigma]E[Z_{n,i,1}|\mu]] && (14)\\
E[B_n] &= CE[Z_{n,i,k}] && (15)\\
E[D_i] &\leq E[D_{i-1}](1-(1-E[Z_{i,0,1}])^{(C)}\Pi_{j=1}^{i-1}(1-E[Z_{i,j,1}])^{E[Y_1]E[S_{1,1}]}) && (16)\\
E[D_1] &= (1-(1-E[Z_{1,0,1}])^{(C)}) && (17)
\end{aligned}$$

The proof is provided in Appendix A. It works by direct calculation and applying Jensen's Inequality to D_i. While ignoring some correlations introduces a nontrivial error into the approximation, this is done to ensure that the expectation can be evaluated in

linear time. Additionally, the tail bounds from Markov's Inequality are relatively loose. This makes it necessary to calibrate t, as simply setting $t = \tau c$ would yield excessively conservative policies. Nevertheless, as we will see in Section 5, this approximation still carries enough information for our policies to work well once tuned.[9]

4.3 Second Moment Policy

First moment policies do not take much information about the structure of deployments into account. In a sense they have to always assume the worst possible population mix and run the risk of accepting deployments with low expected size but high variance when close to the threshold. One way around this is to also take the second moment, i.e., the variance of L_n, into account. To address this, we propose to use Cantelli's inequality, a single-tailed generalization of Chebyshev's inequality, to approximate the probability of scale out failures (i.e., Equality (5)). Cantelli's inequality states that, for a real-valued random variable L and $\epsilon \geq 0$, it holds that

$$Pr(L - E[L] \geq \epsilon) \quad \leq \quad \frac{Var[L]}{Var[L] + \epsilon^2}. \tag{18}$$

If we now set $\epsilon = (c - \sum_{x \in X} E[L_n^x])$, we obtain a bound for the probability of running over capacity that takes more information into account than a first moment policy.

Definition 3 (Second Moment Policy)**.** *Under a second moment policy π with threshold ρ, a newly arriving deployment in belief state b is accepted if, after accepting the deployment, the estimated probability of running over capacity would be less than ρ in all further time steps, i.e.*

$$\sum_{\tilde{x} \in A_\pi(b)} E[L_n^{\tilde{x}}] \leq c \qquad \forall 0 \leq n < N \tag{19}$$

$$\frac{\sum_{\tilde{x} \in A_\pi(b)} Var[L_n^{\tilde{x}}]}{\sum_{\tilde{x} \in A_\pi(b)} Var[L_n^{\tilde{x}}] + (c - \sum_{\tilde{x} \in A_\pi(b)} E[L_n^{\tilde{x}}])^2} \leq \rho \qquad \forall 0 \leq n < N \tag{20}$$

where $E[L_n^{\tilde{x}}]$ is approximated using the approach described in Proposition 2 and $Var[L_n^{\tilde{x}}]$ is approximated using the approach described in Proposition 3.

9. Similar observations have been made in the literature on the use of effective bandwidth for admission control in queueing settings (Kelly, 1991; Berger & Whitt, 1998).

Proposition 3. *Assuming all M_n, D_i, Q_n, and B_n are uncorrelated, it holds:*

$$
\begin{aligned}
V[L_n] &= E[M_n]^2 V[D_n(Q_n + B_n)] + V[M_n] E[D_n(Q_n + B_n)]^2 && (21)\\
&\quad + V[M_n] V[D_n(Q_n + B_n)] && (22)\\
V[D_n(Q_n + B_n)] &= E[D_n]^2 (V[Q_n] + V[B_n]) + (E[Q_n] + E[B_n])^2 V[D_n] && (23)\\
&\quad + V[D_n](V[Q_n] + V[B_n]) && (24)\\
V[Q_n] &= E[V[Q_n|\lambda, \sigma, \mu]] + V[E[Q_n|\lambda, \sigma, \mu]] && (25)\\
V[Q_n|\lambda, \sigma, \mu] &= \sum_{i=1}^{n} \big(\big(V[Y_i|\lambda, \mu] E[S_{i,l}|\sigma]^2 + E[Y_i|\lambda, \mu] V[S_{i,l}|\sigma] \big) E[Z_{n,i,1}|\mu]^2 && (26)\\
&\quad + E[Y_i|\lambda, \mu] E[S_{i,l}|\sigma] V[Z_{n,i,1}|\mu] \big) && (27)\\
V[B_n] &= C E[V[Z_{n,i,k}|\mu]] + C^2 V[E[Z_{n,i,k}|\mu]] && (28)\\
V[D_n] &= E[D_n] - E[D_n]^2 && (29)
\end{aligned}
$$

The proof is provided in Appendix A. It works by direct calculation. Note that, since the expectation is approximated as given in Proposition 2, $V[D_n]$ carries over any approximation errors from $E[D_n]$. As with first moment policies, the bound given by the inequality is again not tight enough to simply set it to $\rho = \tau$ and ρ has to be tuned.

Computational overhead.

The computational overhead of the second moment policy depends on the number of future time steps it evaluates and the chosen prior distributions for the provider's belief state. As long as well-behaved priors are used (e.g., the Gamma priors we use in our simulations), each single rule application is fast. For such priors, updating the estimate for the second moment policy for a single deployment can be done in $O(n)$ where n is the number of evaluated time steps. Whenever a new deployment arrives, the estimate is updated for every active deployment. This leads to a worst case runtime of $O(|X|n)$ where $|X| \leq c$ is the number of active deployments. For multiple clusters this is fully parallelizable at the cluster level because each cluster has its own policy evaluation. Updating the prior of a deployment during runtime has negligible complexity ($O(1)$). A cloud computing center consisting of clusters of capacity c with an arrival rate of L new deployment requests per hour therefore has a computation overhead of at most $O(Lcn)$ each hour, parallelizable into jobs of size $O(n)$.[10] This means that even relatively large look-ahead horizons n can easily be implemented in practice.

5. Empirical Evaluation

In this section, we evaluate the performance of our admission policies using a model fitted to the real-world data trace of Cortez et al. (2017).

10. If further ML is (optionally) employed to obtain an individual prior for arriving deployments (as discussed in Section 6), that computation time would need to be added and depends on the algorithm in question.

Priors	$\mu \approx Gamma(0.3107, 0.5778)$
	$\lambda \approx Gamma(0.4907, 0.4496)$
	$\sigma \approx Gamma(0.2616, 0.0552)$
Global Parameters	$\Delta = 0.119$
	$\nu = 0.673$

Table 1: Fitted processes

5.1 Data Trace and Fitted Model

Cortez et al. (2017) published a data trace consisting of all deployments that populated a Microsoft Azure datacenter in one month. Since the data set is of limited size and only covers one month, we cannot directly evaluate the policies on the historical deployments. One month is too short to fully evaluate cluster admission policies as many effects only show up after months of usage. Instead, we fit processes to the data we do have, to simulate longer time periods (3 years, in our simulations). We defer evaluations against real deployments to future work.

An in-depth discussion of our fitting procedure can be found in Appendix B. The resulting model utilizes Gamma priors, which are a very general distribution (containing the Chi-squared, Erlang and Exponential distributions as special cases) and fit the data well. The fitted parameters are shown in Table 1. The moment approximation resulting from combining Propositions 2 and 3 with these priors is given in Appendix C. In the following we present the results of our simulations.

5.2 Simulation Setup

We simulate clusters with capacity $c = 20,000$ for a 3-year period with all three policies. An average of 1 new deployment per hour arrives according to a Poisson process. The parameters of each arriving deployment are drawn from the fitted distributions presented in Table 1. We tune the threshold for each policy via binary search, subject to meeting an SLA of 0.01%.[11] We verify that the SLA is satisfied on average (over runs and months).

Evaluating our first and second moment policies with a three year time horizon and fine-grained time steps is fast enough to be done in real time. However, doing so would take too much computation power to simulate the thousands of years of cluster operation required for our experiments. Therefore, we use the following approach to simulate clusters with a three-year lifespan with a reasonable number of core-hours. We divide the first and second moment policies into 5 subpolicies and only accept a deployment if all subpolicies accept it. The subpolicies have increasingly fine-grained time steps, but each only evaluates a limited look-ahead horizon: 3 years, 1 year, 1 month, 1 week, and 24 hours. Each subpolicy discretizes its time into 600 timesteps. We performed 500 runs and report the average utilization across all runs. Since failures to scale out are focused in the tail of the runs (e.g., with the tuned zeroth moment parameter only about 1% of runs contain any failures), we employ importance sampling to obtain sufficient samples from the tail to guarantee SLA satisfaction with high confidence. Details about the importance sampling can be found

11. This SLA is somewhat stricter than is typically used in practice, which helps counterbalance our model abstracting away complexities such as fragmentation and node failure.

Policy	Threshold	Utilization
Zeroth Moment	$t = 8,864$	50.45% (48.2, 52.7)
First Moment	$t = 14,223$	66.19% (63.41, 68.94)
Second Moment	$\rho = 0.112$	67.32% (64.35, 70.26)

Table 2: Simulation results showing the performance of the three policies. 95% bootstrap confidence intervals are shown in parentheses.

in Appendix D. To avoid misestimating confidence intervals with biased data, we report 95% bias-corrected and accelerated bootstrap confidence intervals (following (Efron, 1987), 100000 re-samples) instead of standard errors.

5.3 Results

We now compare the utilization of our policies to the industry baseline zeroth moment policy. The results are summarized in Table 2. The zeroth moment policy obtains its best result with a threshold of $t = 8,864$, i.e., new deployments are accepted whenever less than $8,864$ would be active in case of acceptance. This results in an average utilization of 50.45% over the lifetime of the cluster. The first moment policy with threshold $t = 14,223$ increases the utilization by 15.74 percentage points to 66.19%. This constitutes a relative increase in utilization of 31.2% over the zeroth moment policy. Similarly, the second moment policy with threshold $\rho = 0.112$ achieves a utilization of 67.32%, a relative improvement of 33.44%.

At first sight, it may be surprising that the first and second moment policies achieve similar utilization. However, this can be explained as follows. Under both policies, the overwhelming number of simulated clusters never reject a scale out request. However, in a few runs, too many large, long-lived deployments are accepted in the beginning of a cluster's lifetime. This leads to many rejections months or even years in the future. Since this happens early in a cluster's lifetime when not much is known about deployments, the difference between the first and second moment policies is relatively small. This highlights the value of obtaining additional (probabilistic) information about arriving deployments. We study this in the next section.

6. The Value of Deployment-specific Priors

So far, we have assumed that the cluster does not have any information about arriving deployments, except for the initial number of cores. The acceptance decision therefore had to primarily depend on the state of the deployments that are already in the cluster.

Intuitively, a policy could more precisely control whether accepting a deployment would risk violating the SLA if the policy had more information about the future behavior of the specific deployment. One way to obtain such information would be to use machine learning (ML) based on features of the arriving deployment and past deployment patterns of the

submitting user (Cortez et al., 2017). While evaluating particular ML algorithms is beyond the scope of this paper, we evaluate the effect that different levels of available information have. To do this, we need to parameterize the level of knowledge. For this we assume that the cluster simply gets passed some number of observations from each true scaling process distribution of each arriving deployment.[12]

6.1 Improving the Handling of Short-lived Deployments

Our moment policies as defined so far cannot yet make optimal use of this additional prior information. While an optimal policy with good prior information would balance the admission of long-lived and short-lived deployments to keep the utilization more stable over time, the moment policies always accept new deployments on a first-come first-served basis until their constraints are violated. This means that if many very long-lived, slow-scaling deployments arrive in the beginning, the cluster sometimes quickly reaches unsafe belief states in which it stops accepting any new deployments, but for which the critical event lies months or even years in the future. While stopping the admission of new deployments in such a situation is reasonable when no prior information about arriving deployments is available, *with* prior information the policy might know that some arriving deployments will almost surely be dead by the time the cluster has filled up. To make use of this, we now present a heuristic modification of our moment policies such that the resulting policy is allowed to accept deployments that only have a marginal impact on the possible SLA violation, even in unsafe states. As a simple condition for this, we call a deployment *marginal in timestep n* if its expected size is smaller than 10^{-5}, i.e., $E[L_n^x] < 10^{-5}$.

Definition 4 (Marginal Heuristic)**.** *Under a first or second moment policy π with the* marginal heuristic*, a newly arriving deployment x in belief state b is accepted if in each future time step $n < N$, after accepting the deployment, either the underlying moment policy's condition is satisfied or the arriving deployment x is marginal, i.e., $E[L_n^x] < 10^{-5}$.*

Going forward, we use the marginal heuristic, unless explicitly noted. It should be pointed out that this heuristic does not have any effect when the cluster does not have good prior information about arriving deployments. With only the global prior, no deployment is marginal for any future timestep $n < N$.

6.2 Simulation Results

In this section, we present simulation results to demonstrate the value of deployment-specific priors. We simulate the first and second moment policies (with marginal heuristic), now with four different levels of prior information: 0, 1, 5, and 50 observations. Otherwise, we use the same simulation setup as in Section 5. The results are shown in Figure 1.[13] We see that having prior knowledge equivalent to even a single observation improves utilization significantly, resulting in a utilization of 75% and 79.5% for the first and second moment

12. As we have used conjugate prior distributions in our model, this approach matches the standard interpretation of parameters of the posterior distribution in terms of "pseudo observations."

13. We also simulated our policies without the marginal heuristic (see Appendix E), and we observe the same general patterns. As one would expect, without the marginal heuristic, the achieved utilization is somewhat smaller (especially with good priors).

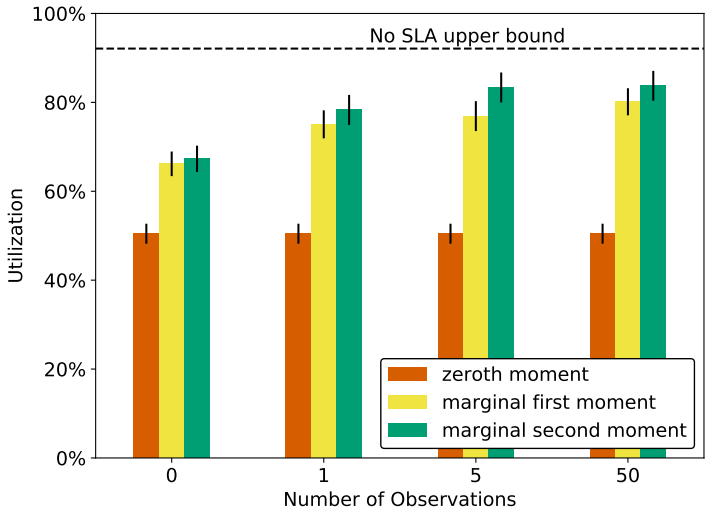

Figure 1: Performance of different policies depending on prior information (error bars indicate 95% bootstrap confidence intervals)

policies, respectively. Better priors lead to even better utilization, with the second moment policy reaching a utilization of 83.8% with 50 observations.

While it is infeasible to calculate the utilization corresponding to an optimal solution of the POMDP, we have derived an *upper bound* of 92.1% by analyzing policies that do not have to satisfy any SLA. Thus, the second moment policy with good prior information achieves more than 90% of the theoretically achievable as given by this (unreachable) upper bound, while delivering a relative increase in utilization of 24.48% above the same policy without prior information and a 66.1% increase over the baseline (i.e., zeroth moment) policy. This shows both the power of our policies and the great importance of taking all available prior information about arriving deployments into account.

7. An Elicitation Mechanism to Improve Priors

Given the importance of the quality of prior information that we established in the last section, in this section, we use techniques from mechanism design to improve this quality. Our approach assumes that users do not typically submit deployments with arbitrary parameters. Instead, they may have a small number of different *types* of deployments. However, the typical mechanism design approach of using a direct revelation mechanism, where customers reveal their full type, seems problematic. First, it may be very cumbersome from a user interface perspective. Second, customers may not have such a detailed understanding of their deployments and thus would run the risk of being penalized for a "misreport." Instead, we seek a design that allows meaningful information to be elicited in an incentive compatible way while being simple for customers to use. To this end, we propose that rather

than explicitly asking users to describe the behavior of their deployments, cloud providers instead provide them with the opportunity to group them into customer-defined categories of roughly similar deployments. Learning priors for each individual category then results in more precise priors and higher utilization. To incentivize such grouping, the cloud provider can set a small portion of the fee for a deployment using a pricing rule based on the variance of resource demands of deployments in a category. We now first present and analyze such a variance-based pricing mechanism and then evaluate the potential utilization gains this mechanism may produce via additional simulations.

7.1 Variance-based Payment Rule

Typically, users are charged a fixed payment per hour for each core their deployment uses. With a variance-based payment rule, we add a small additional charge based on the variance of the estimate for the deployment's scaling process and allow users to *label* the type of their deployments, resulting in an hourly *variance-based payment rule* $q(x)$ of the form:

$$q(x) = \kappa_1 C^x + \kappa_2 Var(x), \tag{30}$$

where κ_1 and κ_2 are price constants and $Var(x)$ is an estimate of the variance of the deployment. A payment rule of this form incentivizes users to assign similar labels to similar deployments to minimize the estimated variance.

To see this, consider a user who has two types of deployment, x and y, with true variances $Var(x)$ and $Var(y)$. He could now simply submit the deployments under a single label. For the provider, this means that each submitted deployment is of either type with a certain probability, which increases the variance of her prediction. But if the user would label each deployment appropriately with either "x" or "y," then the provider would know for each arriving deployment which type it is, reducing variance and therefore the need to reserve capacity. The following proposition, which is immediate from the law of total variance, shows that, at least in the long run, labeling his deployments also reduces a user's payments.

Proposition 4. *Let z be the mixture that results from submitting one of two types of deployments x, y chosen by a Bernoulli random variable $\alpha \sim Bernoulli(p_\alpha)$, i.e., such that z is of type x with probability p_α and of type y with probability $1 - p_\alpha$. Then it holds that*

$$p_\alpha Var(x) + (1 - p_\alpha) Var(y) \leq Var(z) \tag{31}$$

Proof. Since z has finite variance, the law of total variance states:

$$\begin{aligned} Var(z) &= E[Var(z|\alpha)] + Var(E[z|\alpha]) && (32)\\ &\geq E[Var(z|\alpha)] && (33)\\ &= p_\alpha Var(x) + (1 - p_\alpha) Var(y) && (34) \end{aligned}$$

□

Proposition 4 shows that the user would be better off by splitting the mixture and submitting the deployments under separate labels, directly resulting in the following corollary.

Corollary 2. *Under any variance-based payment rule $q(x)$ of the form given in Equation* (30) *with $\kappa_2 > 0$, it is a dominant strategy for users with multiple deployment types to label deployments by type.*

Note that this corollary abstracts away issues of learning and non-stationary strategic behavior; but for reasonable learning procedures we expect a consistent labeling to lead to lower variance than a mixture while learning. Further, this approach not only gives the user correct incentives to reveal the desired information, but actually incentivizes him to improve the performance of the system. In particular, another way he can lower his payment under this scheme (outside the scope of our model) is to design his deployments in such a way that they have lower variance in their resource use. Since more predictable deployments would allow the policy to maintain a smaller buffer, this provides an additional benefit to the system's utilization.

Remark. *The per core-hour payments of a user can be based on* a-priori *or* a-posteriori *estimates of a deployment's variance. With an a-priori estimate, the user knows his payment (per core-hour) before he starts his deployment, which can be very important for certain users. On the other hand, such a payment rule invites strategic deployment submissions: a user could submit a number of small low-variance deployments before submitting a large high-variance deployment, with the goal of reducing the provider's estimate and thus his payment for the large deployment. The provider could mitigate the potential gain of such a manipulation by carefully choosing the estimation procedure, so it is unclear how frequent and successful such manipulations would be in practice. With an a-posteriori estimate (i.e., the user's hourly payment is based on the variance estimate of the deployment at the end of the hour), such strategic deployment submissions could be made unprofitable. However, now users do not know their exact payments in advance. To cater to users that require a fixed price before submitting a deployment, the provider might want to set an upper limit on prices and advertise lower prices as discounts. Thus, which type of estimate is optimal for a given provider depends on the requirements of her user base.*

How much any given user could ultimately save by labeling his deployments mostly depends how different his deployment types are and on how high the provider sets the charge for variance. A user whose deployments are quite uniform will not save much, while a user with some deployments which never scale and some that scale a lot can potentially save a lot. Note that how much the provider should charge is not immediately clear. While she would want to set a high price to put a strong incentive on users, she also has to keep the competition from other providers in mind. At what point the loss of market share outweighs the gain in utilization is an intriguing problem we leave for future work.

7.2 Simulation Results

To illustrate the potential gains in utilization of such a variance-based payment rule, we consider a setting where all users have two deployment types, drawn independently from the same population distribution as in Section 5. We assume that each user only submits a single deployment and then departs, but that the provider has 5 prior observations each from every user's two deployment types. Otherwise, the simulation setup is again the same as in Section 5. In this setting, we contrast the utilization of a provider employing a

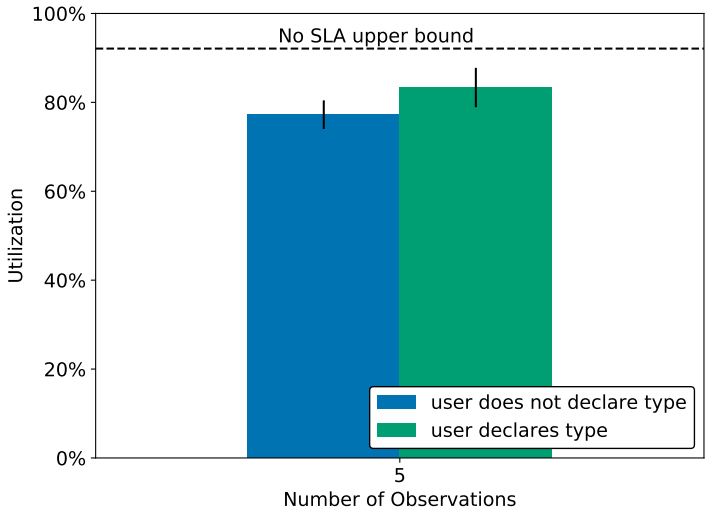

Figure 2: Performance of the second moment policy with two deployment types per user and 5 observations (error bars indicate 95% bootstrap confidence intervals)

second moment policy with and without employing a variance-based payment rule. With the variance-based payment rule (and users consequently declaring their type), the setting becomes equivalent to the one presented in Section 6 with 5 observation. When the variance-based payment rule is *not* used (and users consequently do not label their deployments), we assume that the provider updates her belief for both types independently and evaluates her second moment policy on the mixture.

As we can see in Figure 2, when users do not label their deployments, this yields a utilization of 77%. In contrast, when users do label their deployments, then (as expected) the utilization increases to 83%. This shows that, from a cluster point of view, employing a variance-based payment rule leads to a sizable increase in utilization.

8. Conclusion

We have studied the problem of cluster admission control for cloud computing, where accepting demand now causes unrejectable demand in the future. The optimal policy is given as the solution to a very large constrained POMDP, which is infeasible to solve. In practice, simple threshold policies are employed for admission control. In contrast, we have proposed multiple more sophisticated policies. Our results demonstrate that the utilization can be increased by approximately 30% just from learning about deployments while they are active in the cluster. Furthermore, we have shown that this can be improved to a $50 - 65\%$ gain if better prior information about arriving deployments is available, for example through learning or elicitation techniques. Even though the realized gains in practice are likely to be somewhat lower due to practical engineering constraints (e.g., the need to handle node

outages), they should still be sizable. At cloud scale, even savings of a few percent translate to many hundreds of millions of dollars, and any dollar saved directly translates to a gross profit increase for the cluster provider.

Our work points to a number of interesting future research directions. We have only looked at cluster admission policies at the level of a single cluster, abstracting away the question of which cluster should be chosen, implicitly assuming a first-fit or random-fit heuristic. Future research should look at the question whether filling all clusters with the same mixture of deployments is reasonable or if dedicating different clusters to different types of deployments could be used to further increase utilization. A related direction is that our model and policies assume that deployment behavior does not change during runtime. While this is a reasonable approximation for many deployments, some long running deployments might exhibit more involved life cycles in practice. One way for policies to account for this is to discount past observations.

There are also open questions regarding mechanism design in the cloud domain. In subsequent work, Dierks and Seuken (2020) have already analyzed the competitive effects of employing the variance-based payment rule we proposed in this paper in a duopoly model. They find that, in equilibrium, while using a variance-based payment rule weakly increases welfare, the effects on the providers' profits are ambiguous. However, their analysis does not take into account that a variance-based payment rule can yield better priors about submitted deployments and thus improve the efficiency of the provider's admission policy. Future work could explore an alternative economic design, where the provider offers a menu of two alternatives to her users: the standard alternative, where deployments can always scale out; and a cheaper alternative, where deployments are not allowed to scale out (or can scale only with a "best effort" guarantee). Such an approach could be viewed as implicitly selling finance-style options on the ability to scale out.

References

Abhishek, V., Kash, I. A., & Key, P. (2012). Fixed and market pricing for cloud services. In *7th Workshop on the Economics of Networks, Systems, and Computation (NetEcon)*, pp. 157–162.

Ashlagi, I., Burq, M., Dutta, C., Jaillet, P., Saberi, A., & Sholley, C. (2019). Edge weighted online windowed matching. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 729–742.

Assadi, S., Khanna, S., & Li, Y. (2017). The stochastic matching problem: Beating half with a non-adaptive algorithm. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pp. 99–116.

Azar, Y., Kalp-Shaltiel, I., Lucier, B., Menache, I., Naor, J., & Yaniv, J. (2015). Truthful online scheduling with commitments. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pp. 715–732. ACM.

Babaioff, M., Mansour, Y., Nisan, N., Noti, G., Curino, C., Ganapathy, N., Menache, I., Reingold, O., Tennenholtz, M., & Timnat, E. (2017). Era: a framework for economic resource allocation for the cloud. In *Proceedings of the 26th International Confer-*

ence on World Wide Web Companion, pp. 635–642. International World Wide Web Conferences Steering Committee.

Behnezhad, S., & Reyhani, N. (2018). Almost optimal stochastic weighted matching with few queries. In *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 235–249.

Berger, A. W., & Whitt, W. (1998). Extending the effective bandwidth concept to networks with priority classes. *IEEE Communications magazine*, *36*(8), 78–83.

Brooks, A., Makarenko, A., Williams, S., & Durrant-Whyte, H. (2006). Parametric pomdps for planning in continuous state spaces. *Robotics and Autonomous Systems*, *54*(11), 887–897.

Cohen, M. C., Keller, P. W., Mirrokni, V., & Zadimoghaddam, M. (2019). Overcommitment in cloud services: Bin packing with chance constraints. *Management Science*.

Cortez, E., Bonde, A., Muzio, A., Russinovich, M., Fontoura, M., & Bianchini, R. (2017). Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms. In *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, pp. 153–167, New York, NY, USA. ACM.

Daw, A., & Pender, J. (2018). Queues driven by hawkes processes. *Stochastic Systems*, *8*(3), 192–229.

Delimitrou, C., Bambos, N., & Kozyrakis, C. (2013). Qos-aware admission control in heterogeneous datacenters. In *Proceedings of the 10th International Conference on Autonomic Computing (ICAC 13)*, pp. 291–296, San Jose, CA. USENIX.

Dierks, L., & Seuken, S. (2019). Cloud pricing: The spot market strikes back. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. ACM.

Dierks, L., & Seuken, S. (2020). The competitive effects of variance-based pricing. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pp. 362–370.

Dolev, D., Feitelson, D. G., Halpern, J. Y., Kupferman, R., & Linial, N. (2012). No justified complaints: On fair sharing of multiple resources. In *proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pp. 68–75. ACM.

Duff, M. O., & Barto, A. (2002). *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. Ph.D. thesis, University of Massachusetts at Amherst.

Efron, B. (1987). Better bootstrap confidence intervals. *Journal of the American statistical Association*, *82*(397), 171–185.

Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., & Stoica, I. (2011). Dominant resource fairness: Fair allocation of multiple resource types. In *USENIX Symposium on Networked Systems Design and Implementation*.

Gutman, A., & Nisan, N. (2012). Fair allocation without trade. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pp. 719–728. International Foundation for Autonomous Agents and Multiagent Systems.

Hawkes, A. G. (2018). Hawkes processes and their applications to finance: a review. *Quantitative Finance*, *18*(2), 193–198.

Hindman, B., Konwinski, A., Zaharia, M., Ghodsi, A., Joseph, A. D., Katz, R., Shenker, S., & Stoica, I. (2011). Mesos: A platform for fine-grained resource sharing in the data center. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation*, NSDI'11, pp. 295–308, Berkeley, CA, USA. USENIX Association.

Jyothi, S. A., Curino, C., Menache, I., Narayanamurthy, S. M., Tumanov, A., Yaniv, J., Mavlyutov, R., Goiri, I., Krishnan, S., Kulkarni, J., & Rao, S. (2016). Morpheus: Towards automated slos for enterprise clusters. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 117–134, Savannah, GA. USENIX Association.

Kahn, H., & Marshall, A. W. (1953). Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, *1*(5), 263–278.

Kash, I. A., & Key, P. B. (2016). Pricing the cloud. *IEEE Internet Computing*, *20*(1), 36–43.

Kash, I. A., Procaccia, A. D., & Shah, N. (2014). No agent left behind: Dynamic fair division of multiple resources. *Journal of Artificial Intelligence Research*, *51*, 579–603.

Kelly, F. P. (1991). Effective bandwidths at multi-class queues. *Queueing systems*, *9*(1-2), 5–15.

Khonji, M., Jasour, A., & Williams, B. (2019). Approximability of constant-horizon constrained pomdp. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 5583–5590. International Joint Conferences on Artificial Intelligence Organization.

Kim, D., Lee, J., Kim, K.-E., & Poupart, P. (2011). Point-based value iteration for constrained pomdps. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pp. 1968–1974. AAAI Press.

Lusena, C., Goldsmith, J., & Mundhenk, M. (2001). Nonapproximability results for partially observable markov decision processes. *Journal of artificial intelligence research*, *14*, 83–103.

Ma, W., & Simchi-Levi, D. (2019). Tight weight-dependent competitive ratios for online edge-weighted bipartite matching and beyond. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pp. 727–728.

NIST (2012). Nist/sematech e-handbook of statistical methods. http://www.itl.nist.gov/div898/handbook /apr/section4/apr412.htm.

Papadimitriou, C. H., & Tsitsiklis, J. N. (1987). The complexity of markov decision processes. *Math. Oper. Res.*, *12*(3), 441–450.

Parkes, D. C., Procaccia, A. D., & Shah, N. (2015). Beyond dominant resource fairness: Extensions, limitations, and indivisibilities. *ACM Transactions on Economics and Computation*, *3*(1), 3:1–3:22.

Porta, J. M., Vlassis, N., Spaan, M. T., & Poupart, P. (2006). Point-based value iteration for continuous pomdps. *Journal of Machine Learning Research*, *7*(Nov), 2329–2367.

Poupart, P., Malhotra, A., Pei, P., Kim, K.-E., Goh, B., & Bowling, M. (2015). Approximate linear programming for constrained partially observable markov decision processes. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 3342–3348. AAAI Press.

Rajan, K., Kakadia, D., Curino, C., & Krishnan, S. (2016). Perforator: eloquent performance models for resource optimization. In *Proceedings of the Seventh ACM Symposium on Cloud Computing*, pp. 415–427. ACM.

Roy, N., Gordon, G., & Thrun, S. (2005). Finding approximate pomdp solutions through belief compression. *Journal of artificial intelligence research*, *23*, 1–40.

Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited.

Santana, P., Thiébaux, S., & Williams, B. (2016). Rao*: an algorithm for chance constrained pomdps. In *Proc. AAAI Conference on Artificial Intelligence*.

Schwarzkopf, M., Konwinski, A., Abd-El-Malek, M., & Wilkes, J. (2013). Omega: flexible, scalable schedulers for large compute clusters. In *Proceedings of the 8th ACM European Conference on Computer Systems*, pp. 351–364. ACM.

Smallwood, R. D., & Sondik, E. J. (1973). The optimal control of partially observable markov processes over a finite horizon. *Operations research*, *21*(5), 1071–1088.

Smith, T., & Simmons, R. (2005). Point-based pomdp algorithms: Improved analysis and implementation. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, pp. 542–549, Arlington, Virginia, United States. AUAI Press.

Song, W., Xiao, Z., Chen, Q., & Luo, H. (2014). Adaptive resource provisioning for the cloud using online bin packing. *IEEE Transactions on Computers*, *63*(11), 2647–2660.

Tumanov, A., Zhu, T., Park, J. W., Kozuch, M. A., Harchol-Balter, M., & Ganger, G. R. (2016). Tetrisched: global rescheduling with adaptive plan-ahead in dynamic heterogeneous clusters. In *Proceedings of the Eleventh European Conference on Computer Systems*, p. 35. ACM.

Undurti, A., & How, J. P. (2010). An online algorithm for constrained pomdps. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 3966–3973. IEEE.

Verma, A., Pedrosa, L., Korupolu, M., Oppenheimer, D., Tune, E., & Wilkes, J. (2015). Large-scale cluster management at google with borg. In *Proceedings of the Tenth European Conference on Computer Systems*. ACM.

Walraven, E., & Spaan, M. T. (2018). Column generation algorithms for constrained pomdps. *Journal of Artificial Intelligence Research*, *62*, 489–533.

Wolke, A., Tsend-Ayush, B., Pfeiffer, C., & Bichler, M. (2015). More than bin packing. *Inf. Syst.*, *52*(C), 83–95.

Yan, Y., Gao, Y., Chen, Y., Guo, Z., Chen, B., & Moscibroda, T. (2016). Tr-spark: Transient computing for big data analytics. In *SoCC*.

Zhao, J., Yang, K., Wei, X., Ding, Y., Hu, L., & Xu, G. (2016). A heuristic clustering-based task deployment approach for load balancing using bayes theorem in cloud environment. *IEEE Transactions on Parallel and Distributed Systems*, *27*(2), 305–316.

Appendix A. Omitted Proofs

Proof of Proposition 2. • If M_n, D_n, Q_n and B_n are uncorrelated, it holds by linearity and multiplicativity of the expected value for uncorrelated random variables:

$$E[L_n] = E[M_n]E[D_n](E[Q_n] + E[B_n]) \tag{35}$$

- Q_n: For the expectation of Q_n it holds:

$$\begin{aligned}
E[Q_n] &= E[\sum_{i=1}^{n} \sum_{k=1}^{\sum_{l=0}^{Y_i} S_{i,l}} Z_{n,i,k}] && (36)\\
&= \sum_{i=1}^{n} E[\sum_{k=1}^{\sum_{l=0}^{Y_i} S_{i,l}} Z_{n,i,k}] && (37)\\
&= \sum_{i=1}^{n} E[E[\sum_{k=1}^{\sum_{l=0}^{Y} S_{i,l}} Z_{n,i,k}|\lambda,\sigma,\mu]] && (38)\\
E[\sum_{k=1}^{\sum_{l=0}^{Y} S_{i,l}} Z_{n,i,k}|\lambda,\sigma,\mu] &= E[E[\sum_{k=1}^{\sum_{l=0}^{Y} S_{i,l}} Z_{n,i,k}|\sum_{l=0}^{Y_i} S_{i,l}]|\lambda,\sigma,\mu] && (39)\\
&= E[\sum_{l=0}^{Y_i} S_{i,l}|\lambda,\sigma,\mu]E[Z_{n,i,1}|\lambda,\sigma,\mu] && (40)\\
&= E[Y_1|\lambda,\mu]E[S_{1,1}|\sigma]E[Z_{n,i,1}|\mu] && (41)
\end{aligned}$$

- B_n: By definition, it holds

$$E[B_n] = E[\sum_{j=1}^{C} Z_{n,0,k}] = CE[Z_{n,0,k}] \tag{42}$$

- D_i: If all $Z_{i,j,k}$, Y_1 and $S_{i,l}$ are uncorrelated, it holds

$$\begin{aligned}
E[D_i] &= E[D_{i-1}(1 - \Pi_{j=0}^{i-1}\Pi_{k=0}^{\sum_{l=0}^{Y_j} S_{j,l}}(1 - Z_{i,j,k}))] && (43)\\
&= E[D_{i-1}](1 - E[\Pi_{j=0}^{i-1}\Pi_{k=0}^{\sum_{l=0}^{Y_j} S_{j,l}}(1 - Z_{i,j,k})]) && (44)\\
&= E[D_{i-1}](1 - E[E[\Pi_{j=0}^{i-1}\Pi_{k=0}^{\sum_{l=0}^{Y_j} S_{j,l}}(1 - Z_{i,j,k})|Y,S]]) && (45)\\
&= E[D_{i-1}](1 - E[\Pi_{j=0}^{i-1}\Pi_{k=0}^{\sum_{l=0}^{Y_j} S_{j,l}}(1 - E[Z_{i,j,k}])]) && (46)\\
&= E[D_{i-1}](1 - E[\Pi_{j=0}^{i-1}(1 - E[Z_{i,j,k}])^{\sum_{l=0}^{Y_j} S_{j,l}}]) && (47)\\
&\leq E[D_{i-1}](1 - \Pi_{j=0}^{i-1}(1 - E[Z_{i,j,k}])^{E[\sum_{l=0}^{Y_j} S_{j,l}]}) && (48)\\
&= E[D_{i-1}](1 - \Pi_{j=0}^{i-1}(1 - E[Z_{i,j,k}])^{E[Y_1]E[S_{1,1}]}) && (49)\\
E[D_1] &= (1 - (1 - E[Z_{1,0,1}])^{C}) && (50)
\end{aligned}$$

where the third line follows by the law of total probability and the 6'th by Jensens Inequality.

□

Proof of Proposition 3. With M_n,D_n,Q_n and B_n uncorrelated, it holds for the variance of L_n:

$$\begin{aligned} V[L_n] &= V[M_n D_n (Q_n + B_n)] && (51)\\ &= E[M_n]^2 V[D_n(Q_n + B_n)] + V[M_n] E[D_n(Q_n + B_n)]^2 && (52)\\ &\quad + V[M_n] V[D_n(Q_n + B_n)] && (53)\\ & && (54) \end{aligned}$$

and further:

$$\begin{aligned} V[D_n(Q_n + B_n)] &= E[D_n]^2 (V[Q_n] + V[B_n]) + (E[Q_n] + E[B_n])^2 V[D_n] && (55)\\ &\quad + V[D_n](V[Q_n] + V[B_n]) && (56)\\ & && (57) \end{aligned}$$

For the variance of Q_n it holds:

$$V[Q_n] = E[V[Q_n|\lambda,\sigma,\mu]] + V[E[Q_n|\lambda,\sigma,\mu]] \quad (58)$$

and

$$V[Q_n|\lambda,\sigma,\mu] = \sum_{i=1}^{n} \left(V[\sum_{l=0}^{Y_i} S_{i,l}|\lambda,\mu,\sigma] E[Z_{n,i,1}|\mu]^2 + E[\sum_{l=0}^{Y_i} S_{i,l}|\lambda,\mu,\sigma] V[Z_{n,i,1}|\mu] \right) \quad (59)$$

$$= \sum_{i=1}^{n} \left((V[Y_i] E[S_{i,l}]^2 + E[Y_i] V[S_{i,l}]) E[Z_{n,i,1}]^2 + E[Y_i] E[S_{i,l}] V[Z_{n,i,1}] \right) \quad (60)$$

by the law of total variance.

For B_n we can now use the law of total variance to obtain:

$$\begin{aligned} V[B_n] &= V[\sum_{j=1}^{C} Z_{n,i,k}] && (61)\\ &= E[V[\sum_{j=1}^{C} Z_{n,i,k}|\mu]] + V[E[\sum_{j=1}^{C} Z_{n,i,k}|\mu]] && (62)\\ &= E[CV[Z_{n,i,k}|\mu]] + V[CE[Z_{n,i,k}|\mu]] && (63)\\ &= CE[V[Z_{n,i,k}|\mu]] + C^2 V[E[Z_{n,i,k}|\mu]] && (64) \end{aligned}$$

Lastly, for D_n, note that $E[D_n^2] = E[D_n]$ because $D_n \in \{0,1\}$. It follows

$$V[D_n] = E[D_n] - E[D_n]^2 \quad (65)$$

□

Appendix B. Data Trace

To have a better understanding of the scaling behavior of real deployments and to create a model suitable for simulating clusters, we fitted the behavior of deployments to a real-world data trace. The particular data trace we use was published by Cortez et al. (2017). This dataset consists of one month of data of internal Microsoft Azure jobs. It contains 35,576 deployments,[14] though only 29,757 of these deployments arrived during the observed time period. Since we want to fit distributions with the goal of simulating arriving deployments, only these 29,757 deployments can be used for most of our fitting. The deployments that arrived before the beginning of the observed period of time cannot be used when making maximum likelihood estimations, because for start times before the observed period of time, only longer lived deployments survived to be observed. Including them would strongly skew the fit. The 29,757 deployments activated 4,317,961 cores, out of which 4,211,926 became inactive again during the observed month. The exact lifetime of the remaining cores (i.e., the length of time between becoming active and then inactive again) is not known; instead we only have a lower bound on it (i.e., our observation is Type I censored: see for example (NIST, 2012)). Thus, for cores where we only have a lower bound on the lifetime we use the cdf in our likelihood function while for cores whose lifetime is known we use the pdf.

B.1 Fitting on the Deployment Level

We first fit arrival and departure processes for each individual deployment. In keeping with the Markov assumption, we fit a Poisson distribution to the scale out rate of each deployment, while we fit an exponential distribution to the lifetime of cores for each deployment for which at least one core became inactive during the observed time period. Note that while we model the cluster admission problem as a discrete-time POMDP, the processes are fit in continuous time. This is more general and avoids imprecisions introduced by time discretization. To fit the size of a scale out, we also used a Poisson distribution (plus 1, as scale outs must have at least one core).[15] We further assume that each deployment, had it lived forever, at some point would have made a scale out request for more than 1 core. Since we did not observe these scale-outs and therefore cannot make a direct likelihood fit, we introduce two parameters P_1 and P_2 to represent them. We assume that the scale out rate of deployments that never scaled out (some because they died, but many simply because the observation period of the dataset ended) is equal to the value for which not observing a scale out has probability P_1. We equivalently set the scale out size for deployments that never increase their size by more than 1 core during one scale out event according to P_2. We calibrated P_1 and P_2 by minimizing the (discrete) Cramér-von Mises distance of the size of deployments between samples drawn from our fitted model and the data set. The optimal distance is 0.1585 and an overlay of both cumulative distribution functions can be seen in Figure 3. Note that most of the remaining distance does not seem to be caused

14. In contrast to (Cortez et al., 2017) we did not consolidate all deployments a single user runs on a certain day into one. This is because cores that get requested as a new deployment do not need to be accepted on the same cluster.

15. As the Poisson distribution is single-parameter and its variance cannot be set independent of the average size, this is not a particularly good fit for users with large but consistent scale out sizes. However, its simplicity avoids overfitting on the often low number of samples per deployment and it results in a good fit on the population level.

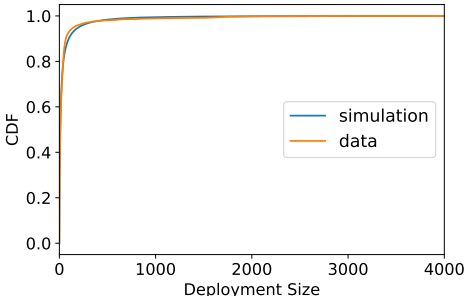

Figure 3: CDF over number of deployments of all sizes

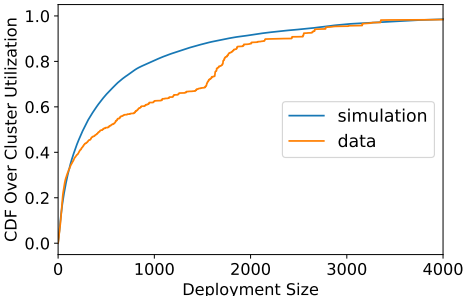

Figure 4: CDF over utilization percentage from deployments of all sizes

by limitations of our model or fitting procedure, but by limitations of the dataset. The dataset, while relatively large, still does contain a somewhat small selection of deployments from the tail. More importantly, it only contains *internal* Azure deployments, so the types of workloads are limited. As such, it contains few deployments of sizes between 100 and 1500, but a relatively large number of deployments of sizes between 1500 and 2000. This effect is visualized in Figure 4, which shows the CDF over the percentage of utilization in the cluster coming from deployments of different sizes for both our model and the dataset.

B.2 Fitting on the Population Level

With the distributions for each deployment in place, we now fit Gamma distributions for the population. The parameters of the processes for each arriving deployment are drawn from these populations. As the data was skewed, positive, and not really heavy tailed, a Gamma distribution is a natural and very general candidate (containing the Chi-squared, Erlang and Exponential distributions as special cases), with the added benefit of being conjugate prior to the deployment processes. The resulting model and parameters from our fits are shown in Table 3. While the scale out size is fit directly to the samples, scale out rate and core lifetime are highly correlated. The longer a deployment's cores live, the lower the rate at which new cores arrive, as can be seen in Figure 5. This shows that deployments with long lived cores do not necessarily have more active cores. To account for this, we fit the power law relationship ν between scale out rate and lifetime, i.e., we fitted the prior distribution on scale out rates multiplied by the respective core lifetimes taken to the power

Priors	$\mu \approx Gamma(0.3107, 0.5778)$ $\lambda \approx Gamma(0.4907, 0.4496)$ $\sigma \approx Gamma(0.2616, 0.0552)$
Global Parameters	$\Delta = 0.119$ $\nu = 0.673$

Table 3: Fitted processes

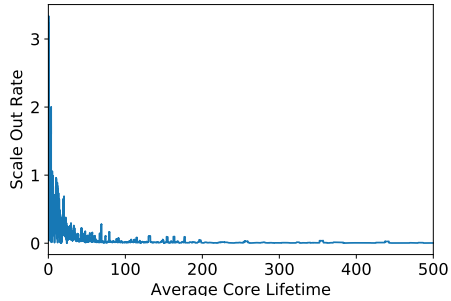

Figure 5: Scale out rate as a function of average core lifetime

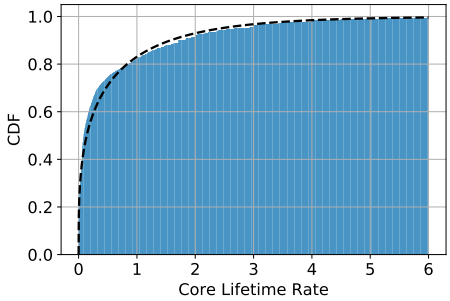

Figure 6: Distribution of the core lifetime rate parameter

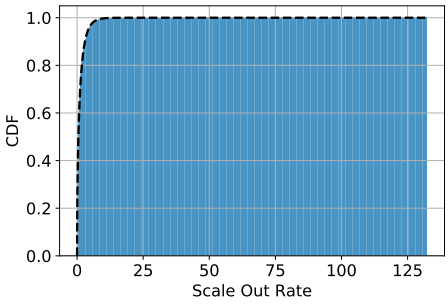

Figure 7: Distribution of the scale out rate parameter

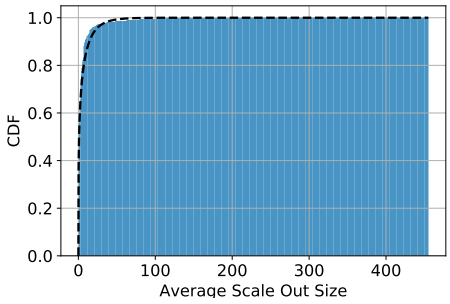

Figure 8: Distribution of the scale out size parameter

of ν. We have chosen ν such that the mean absolute distance between normalized scale out rate of each deployment and the average (normalized) scale out rate is minimized.

To visualize the fitted distributions, Figure 6 shows the CDF of the Gamma distribution for the lifetime parameter, overlaid over the normalized cumulative histogram of the fitted rates of the sample deployments.

Figure 7 shows the CDF for the normalized scale out rates over the relevant cumulative histogram. The actual scale out rate of a sampled deployment is now simply the normalized scale out rate multiplied by the average core lifetime. Figure 8 shows the fitted CDF for the scale out size parameter over its cumulative histogram.

Deployment Shutdown. While most deployments in the dataset die because they have zero active cores, 5,980 of the 22,241 deployments that both arrive and die during the observed period seem to get actively shut down. By this we mean that they had at least 3 VMs that all shut down simultaneously. This would be highly unlikely if deployments only die when cores or VMs become inactive independently. To capture such behavior we fit an exponential distribution over the number of expected core lifetime deployments lived. The

maximal lifetime of deployments that did not get shut down was assumed to be censored to their realized lifetime.

Appendix C. Moment Approximation with Gamma Priors

Proposition 5. *When* $Y_i \sim Pois(\lambda\mu^\nu)$, $\lambda \sim Gamma(a,b)$, $S_{i,l} \sim Pois(\sigma)$, $\sigma \sim Gamma(\alpha,\beta)$, $Z_{n,i,j} \sim Bernoulli(e^{(i-n)\mu})$ *(Bernoulli over complementary CDF of an exponential distribution),* $\mu \sim Gamma(\mathfrak{a},\mathfrak{b})$ *and* $M_i \sim Bernoulli(e^{(i-n)\Delta\mu})$ *it holds:*

$$E[Q_n] = \frac{a}{b}\frac{\alpha+\beta}{\beta}\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})}\sum_{i=1}^{n}\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}} \tag{66}$$

$$E[D_i] \leq E[D_{i-1}](1-\Pi_{j=0}^{i-1}(1-(1+\frac{i-j}{\mathfrak{b}})^{-(\mathfrak{a})})^{\frac{a}{b}\frac{\alpha}{\beta}}) \tag{67}$$

$$E[Z_{n,i,1}] = \frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}} \tag{68}$$

$$E[M_n] = \frac{\mathfrak{b}^{\mathfrak{a}}}{(\Delta n+\mathfrak{b})^{\mathfrak{a}}} \tag{69}$$

and

$$V[Q_n] = \mathfrak{b}^{\mathfrak{a}}\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})}\left[\left(\frac{a}{b}(\frac{\alpha}{\beta^2}+\frac{\alpha+\beta}{\beta}^2-1)\right)\right. \tag{70}$$

$$\left.\sum_{i=1}^{n}\frac{1}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}+\nu}}+\frac{a}{b}\frac{\alpha+\beta}{\beta}\sum_{i=1}^{n}\frac{1}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}}\right] \tag{71}$$

$$+\left(\frac{a}{b}^2\frac{\alpha+\beta}{\beta}^2+(\frac{a}{b}^2\frac{\alpha}{\beta^2}+\frac{a}{b^2}\frac{\alpha+\beta}{\beta}^2+\frac{a}{b^2}\frac{\alpha}{\beta^2})\right) \tag{72}$$

$$\left[\mathfrak{b}^{\mathfrak{a}}\frac{\Gamma(\mathfrak{a}+2\nu)}{\Gamma(\mathfrak{a})}\sum_{1\leq i\leq j\leq n}\frac{1}{(2n+\mathfrak{b}-i-j)^{\mathfrak{a}+2\nu}}\right. \tag{73}$$

$$\left.-(\mathfrak{b}^{\mathfrak{a}}\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})})^2\sum_{1\leq i\leq j\leq n}\frac{1}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}}\frac{1}{(n+\mathfrak{b}-j)^{\mathfrak{a}+\nu}}\right] \tag{74}$$

$$+(\frac{a}{b}^2\frac{\alpha}{\beta^2}+\frac{a}{b^2}\frac{\alpha+\beta}{\beta}^2+\frac{a}{b^2}\frac{\alpha}{\beta^2})\left[\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})}\sum_{i=1}^{n}\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}}\right]^2 \tag{75}$$

$$V[Z_{n,i,1}] = \frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}}(1-\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}}) \tag{76}$$

$$CE[V[Z_{n,i,k}|\mu]]+C^2V[E[Z_{n,i,k}|\mu]] = C\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}}(1-C\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}}) \tag{77}$$

$$+(C^2-C)\frac{\mathfrak{b}^{\mathfrak{a}}}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}}} \tag{78}$$

$$V[M_n] = \frac{\mathfrak{b}^{\mathfrak{a}}}{(\Delta n+\mathfrak{b})^{\mathfrak{a}}}(1-\frac{\mathfrak{b}^{\mathfrak{a}}}{(\Delta n+\mathfrak{b})^{\mathfrak{a}}}) \tag{79}$$

Proof. • For Q_n it holds

$$\begin{aligned} E[Q_n] &= \sum_{i=1}^{n} E[E[Y_1|\lambda,\mu]E[S_{1,1}|\sigma]E[Z_{n,i,1}|\mu]] && (80)\\ &= \sum_{i=1}^{n} E[Y_1|\lambda,\mu]E[S_{1,1}|\sigma]E[Z_{n,i,1}|\mu] && (81)\\ &= \sum_{i=1}^{n} \lambda\mu^{\nu}(\sigma+1)e^{(i-n)\mu} && (82) \end{aligned}$$

$$\begin{aligned} E[\lambda] &= \frac{a}{b} && (83)\\ V[\lambda] &= \frac{a}{b^2} && (84)\\ E[\sigma+1] &= \frac{\alpha+\beta}{\beta} && (85)\\ V[\sigma+1] &= \frac{\alpha}{\beta^2} && (86)\\ E[\mu^{\nu}e^{(i-n)\mu}] &= \int_0^{\infty} \mu^{\nu}e^{(i-n)\mu}\frac{\mathfrak{b}^{\mathfrak{a}}\mu^{\mathfrak{a}-1}e^{-\mathfrak{b}\mu}}{\Gamma(\mathfrak{a})}d\mu && (87)\\ &= \frac{\mathfrak{b}^{\mathfrak{a}}}{\Gamma(\mathfrak{a})}\int_0^{\infty} \mu^{\mathfrak{a}-1+\nu}e^{(i-n-\mathfrak{b})\mu}d\mu && (88)\\ &= \frac{\mathfrak{b}^{\mathfrak{a}}}{\Gamma(\mathfrak{a})}(n+\mathfrak{b}-i)^{-\mathfrak{a}-\nu}\Gamma(\mathfrak{a}+\nu) && (89)\\ &= \frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}}\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})} && (90) \end{aligned}$$

It immediately follows:

$$E[Q_n] = \frac{a}{b}\frac{\alpha+\beta}{\beta}\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})}\sum_{i=1}^{n}\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}} \quad (91)$$

Next we will calculate

$$\begin{aligned} V[Q_n] &= E[V[Q_n|\lambda,\sigma,\mu]] && (92)\\ &\quad +V[E[Q_n|\lambda,\sigma,\mu]] && (93)\\ & && (94) \end{aligned}$$

Before we can do so, we need to collect a few easy supporting results:

$$\begin{aligned} V[Y_1|\lambda,\mu] &= \lambda\mu^{\nu} && (95)\\ V[S_{1,1}|\sigma] &= \sigma && (96)\\ E[Z_{n,i,1}|\mu]^2 &= e^{((i-n)\mu)^2} && (97) \end{aligned}$$

$$
\begin{aligned}
&= e^{(2i-2n)\mu} && (98)\\
&= E[Z_{2n,2i,1}|\mu] && (99)\\
V[Z_{n,i,1}|\mu] &= e^{((i-n)\mu)}(1-e^{((i-n)\mu)}) && (100)\\
&= e^{((i-n)\mu)} - e^{2((i-n)\mu)} && (101)\\
&= E[Z_{n,i,1}|\mu] - E[Z_{2n,2i,1}|\mu] && (102)\\
& && (103)
\end{aligned}
$$

We also need

$$
\begin{aligned}
E[\lambda^2] &= V[\lambda] + E[\lambda]^2 && (104)\\
&= \frac{a}{b^2} + \frac{a}{b}^2 && (105)\\
E[(\sigma+1)^2] &= V[\sigma+1] + E[\sigma+1]^2 && (106)\\
&= \frac{\alpha}{\beta^2} + \frac{\alpha+\beta}{\beta}^2 && (107)\\
E[\mu^{2\nu}e^{(2i-2n)\mu}] &= \int_0^\infty \mu^{2\nu}e^{(2i-2n)\mu}\frac{\mathfrak{b}^{\mathfrak{a}}\mu^{\mathfrak{a}-1}e^{-\mathfrak{b}\mu}}{\Gamma(\mathfrak{a})}d\mu && (108)\\
&= \frac{\mathfrak{b}^{\mathfrak{a}}}{\Gamma(\mathfrak{a})}\int_0^\infty \mu^{\mathfrak{a}-1+2\nu}e^{(2i-2n-\mathfrak{b})\mu}d\mu && (109)\\
&= \frac{\mathfrak{b}^{\mathfrak{a}}}{\Gamma(\mathfrak{a})}(2n+\mathfrak{b}-2i)^{-\mathfrak{a}-2\nu}\Gamma(\mathfrak{a}+2\nu) && (110)\\
&= \frac{\Gamma(\mathfrak{a}+2\nu)}{\Gamma(\mathfrak{a})}\frac{\mathfrak{b}^{\mathfrak{a}}}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}+2\nu}} && (111)
\end{aligned}
$$

This now allows us to calculate everything that is needed for the first half of the variance of Q_n, i.e., $E[V[Q_n|\lambda,\sigma,\mu]]$. First note that

$$
\begin{aligned}
V[Q_n|\lambda,\sigma,\mu] &= \sum_{i=1}^{n}(\big(V[Y_i|\lambda,\mu]E[S_{i,l}|\sigma]^2 && (112)\\
&\quad + E[Y_i|\lambda,\mu]V[S_{i,l}|\mu])\,E[Z_{n,i,1}|\mu]^2 && (113)\\
&\quad + E[Y_i|\lambda]E[S_{i,l}|\sigma]V[Z_{n,i,1}|\mu]) && (114)
\end{aligned}
$$

and

$$
\begin{aligned}
E[(V[Y_i|\lambda,\mu]E[S_{i,l}|\sigma]^2E[Z_{n,i,1}|\mu]^2] &= E[(\lambda)(\sigma+1)^2\mu^\nu e^{(2i-2n)\mu}] && (115)\\
&= E[\lambda]E[(\sigma+1)^2]E[\mu^\nu e^{(2i-2n)\mu}] && (116)\\
&= \frac{a}{b}(\frac{\alpha}{\beta^2} + \frac{\alpha+\beta}{\beta}^2)\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})}\frac{\mathfrak{b}^{\mathfrak{a}}}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}+\nu}} && (117)
\end{aligned}
$$

$$
E[E[Y_i|\lambda,\mu]V[S_{i,l}|\mu])E[Z_{n,i,1}|\mu]^2] = E[\lambda\mu^\nu\sigma e^{(2i-2n)\mu}] \quad (118)
$$

$$= E[\lambda]E[\sigma]E[\mu^\nu e^{(2i-2n)\mu}] \quad (119)$$
$$= \frac{a}{b}\frac{\alpha}{\beta}\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})}\frac{\mathfrak{b}^\mathfrak{a}}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}+\nu}} \quad (120)$$

and

$$E[E[Y_i|\lambda,\mu]E[S_{i,l}|\sigma]V[Z_{n,i,1}|\mu]] = E[\lambda\mu^\nu(\sigma+1)(E[Z_{n,i,1}|\mu] - E[Z_{2n,2i,1}|\mu])] \quad (121)$$
$$= E[\lambda\mu^\nu(\sigma+1)(e^{(i-n)\mu} - e^{(2i-2n)\mu})] \quad (122)$$
$$= E[\lambda]E[\sigma+1](\mu^\nu E[e^{(i-n)\mu}] - E[\mu^\nu e^{(2i-2n)\mu}]) \quad (123)$$
$$= \frac{a}{b}\frac{\alpha+\beta}{\beta}\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})}\left(\frac{\mathfrak{b}^\mathfrak{a}}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}} - \frac{\mathfrak{b}^\mathfrak{a}}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}+\nu}}\right) \quad (124)$$

It follows:

$$E[(V[Y_i|\lambda,\mu]E[S_{i,l}|\sigma]^2E[Z_{n,i,1}|\mu]^2] = \mathfrak{b}^\mathfrak{a}\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})}\left[\left(\frac{a}{b}(\frac{\alpha}{\beta^2} + \frac{\alpha+\beta}{\beta}^2) + \frac{a}{b}\frac{\alpha}{\beta} - \frac{a}{b}\frac{\alpha+\beta}{\beta}\right)\right. \quad (125)$$
$$\left.\frac{1}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}+\nu}} + \frac{a}{b}\frac{\alpha+\beta}{\beta}\frac{1}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}}\right] \quad (126)$$
$$= \mathfrak{b}^\mathfrak{a}\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})}\left[\left(\frac{a}{b}(\frac{\alpha}{\beta^2} + \frac{\alpha+\beta}{\beta}^2 - 1)\right)\right. \quad (127)$$
$$\left.\frac{1}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}+\nu}} + \frac{a}{b}\frac{\alpha+\beta}{\beta}\frac{1}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}}\right] \quad (128)$$

Finally for the second part of the variance, i.e., $V[E[Q_n|\lambda,\sigma,\mu]]$, we need:

$$V[\sum_{i=1}^{n}\mu^\nu e^{(i-n)\mu}] = \sum_{1\le i\le j\le n} Cov[\mu^\nu e^{(i-n)\mu}, \mu^\nu e^{(j-n)\mu}] \quad (129)$$
$$= \sum_{1\le i\le j\le n} E[\mu^\nu e^{(i-n)\mu}\mu^\nu e^{(j-n)\mu}] - E[\mu^\nu e^{(i-n)\mu}]E[\mu^\nu e^{(j-n)\mu}] \quad (130)$$
$$= \sum_{1\le i\le j\le n} E[\mu^{2\nu} e^{(i+j-2n)\mu}] - E[\mu^\nu e^{(i-n)\mu}]E[\mu^\nu e^{(j-n)\mu}] \quad (131)$$
$$= \sum_{1\le i\le j\le n} (\frac{\mathfrak{b}^\mathfrak{a}}{(2n+\mathfrak{b}-i-j)^{\mathfrak{a}+2\nu}}\frac{\Gamma(\mathfrak{a}+2\nu)}{\Gamma(\mathfrak{a})} \quad (132)$$
$$- \frac{(\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})})^2\frac{\mathfrak{b}^\mathfrak{a}}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}}\frac{\mathfrak{b}^\mathfrak{a}}{(n+\mathfrak{b}-j)^{\mathfrak{a}+\nu}}) \quad (133)$$
$$= \mathfrak{b}^\mathfrak{a}\frac{\Gamma(\mathfrak{a}+2\nu)}{\Gamma(\mathfrak{a})}\sum_{1\le i\le j\le n}\frac{1}{(2n+\mathfrak{b}-i-j)^{\mathfrak{a}+2\nu}} \quad (134)$$

$$- (\mathfrak{b}^{\mathfrak{a}} \frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})})^2 \sum_{1 \leq i \leq j \leq n} \frac{1}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}} \frac{1}{(n+\mathfrak{b}-j)^{\mathfrak{a}+\nu}} \tag{135}$$

$$\begin{aligned} V[\lambda(\sigma+1)] &= E[\lambda]^2 V[\sigma+1] + V[\lambda] E[\sigma+1]^2 + V[\lambda] V[\sigma+1] && (136) \\ &= \frac{a}{b}^2 \frac{\alpha}{\beta^2} + \frac{a}{b^2} \frac{\alpha+\beta}{\beta}^2 + \frac{a}{b^2} \frac{\alpha}{\beta^2} && (137) \end{aligned}$$

Now we can write:

$$\begin{aligned} V[E[Q_n|\lambda,\sigma,\mu]] =& V[\sum_{i=1}^{n} E[\sum_{k=1}^{\sum_{l=0}^{Y} S_{i,l}} Z_{n,i,k}|\lambda,\sigma,\mu]] && (138) \\ =& V[\sum_{i=1}^{n} \lambda \mu^{\nu} (\sigma+1) e^{(i-n)\mu}] && (139) \\ =& V[\lambda(\sigma+1) \sum_{i=1}^{n} \mu^{\nu} e^{(i-n)\mu}] && (140) \\ =& E[\lambda(\sigma+1)]^2 V[\sum_{i=1}^{n} \mu^{\nu} e^{(i-n)\mu}] + V[\lambda(\sigma+1)] V[\sum_{i=1}^{n} \mu^{\nu} e^{(i-n)\mu}] && (141) \\ & + V[\lambda(\sigma+1)] E[\sum_{i=1}^{n} \mu^{\nu} e^{(i-n)\mu}]^2 && (142) \\ =& \left(\frac{a}{b}^2 \frac{\alpha+\beta}{\beta}^2 + (\frac{a}{b}^2 \frac{\alpha}{\beta^2} + \frac{a}{b^2} \frac{\alpha+\beta}{\beta}^2 + \frac{a}{b^2} \frac{\alpha}{\beta^2}) \right) && (143) \\ & \left(\mathfrak{b}^{\mathfrak{a}} \frac{\Gamma(\mathfrak{a}+2\nu)}{\Gamma(\mathfrak{a})} \sum_{1 \leq i \leq j \leq n} \frac{1}{(2n+\mathfrak{b}-i-j)^{\mathfrak{a}+2\nu}} \right. && (144) \\ & \left. - (\mathfrak{b}^{\mathfrak{a}} \frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})})^2 \sum_{1 \leq i \leq j \leq n} \frac{1}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}} \frac{1}{(n+\mathfrak{b}-j)^{\mathfrak{a}+\nu}} \right) && (145) \\ & + (\frac{a}{b}^2 \frac{\alpha}{\beta^2} + \frac{a}{b^2} \frac{\alpha+\beta}{\beta}^2 + \frac{a}{b^2} \frac{\alpha}{\beta^2}) \left(\frac{\Gamma(\mathfrak{a}+\nu)}{\Gamma(\mathfrak{a})} \sum_{i=1}^{n} \frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}+\nu}} \right)^2 && (146) \end{aligned}$$

Inserting into Propositions 1 and 2 now yields the result.

- For D_i note the following: As an exponential distribution whose rate is drawn from a Gamma distribution with shape $\mathfrak{a}$ and rate $\mathfrak{b}$ is equal to a Lomax distribution with scale $\mathfrak{b}$ and shape $\mathfrak{a}$, a single $Z_{i,j,k}$ is equal to a Bernoulli trial over the complementary CDF of the Lomax distribution.

$$\begin{aligned} E[Z_{i,j,k}] &= (1 + \frac{i-j}{\mathfrak{b}})^{-(\mathfrak{a})} && (147) \\ & && (148) \end{aligned}$$

It therefore holds

$$E[D_i] \leq E[D_{i-1}](1-\Pi_{j=0}^{i-1}(1-(1+\frac{i-j}{\mathfrak{b}})^{-(\mathfrak{a})})^{\frac{a}{b}\frac{\alpha}{\beta}}) \quad (149)$$

$$E[D_1] = (1-(1-(1+\frac{1}{\mathfrak{b}})^{-(\mathfrak{a})})^{C}) \quad (150)$$

- It now directly follows

$$V[Z_{n,i,1}] = E[V[Z_{n,i,1}|\mu]] + V[E[Z_{n,i,1}|\mu]] \quad (151)$$

$$= E[E[Z_{n,i,1}|\mu]] - E[E[Z_{2n,2i,1}|\mu]] + V[e^{(i-n)\mu}] \quad (152)$$

$$= \frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}} - \frac{\mathfrak{b}^{\mathfrak{a}}}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}}} \quad (153)$$

$$+\frac{\mathfrak{b}^{\mathfrak{a}}}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}}} - \frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}}\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}} \quad (154)$$

$$= \frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}}(1-\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}}) \quad (155)$$

$$CE[V[Z_{n,i,k}|\mu]] + C^2V[E[Z_{n,i,k}|\mu]] \quad (156)$$

$$=C(E[E[Z_{n,i,1}|\mu]] - E[E[Z_{2n,2i,1}|\mu]]) + C^2V[e^{(i-n)\mu}] \quad (157)$$

$$=C(\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}} - \frac{\mathfrak{b}^{\mathfrak{a}}}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}}}) + C^2(\frac{\mathfrak{b}^{\mathfrak{a}}}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}}} - (\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}})^2) \quad (158)$$

$$=C\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}}(1-C\frac{\mathfrak{b}^{\mathfrak{a}}}{(n+\mathfrak{b}-i)^{\mathfrak{a}}}) + (C^2-C)\frac{\mathfrak{b}^{\mathfrak{a}}}{(2n+\mathfrak{b}-2i)^{\mathfrak{a}}} \quad (159)$$

- For M_i it holds by the same argument,

$$E[M_n] = \frac{\mathfrak{b}^{\mathfrak{a}}}{(\Delta n+\mathfrak{b})^{\mathfrak{a}}} \quad (160)$$

$$V[M_n] = \frac{\mathfrak{b}^{\mathfrak{a}}}{(\Delta n+\mathfrak{b})^{\mathfrak{a}}}(1-\frac{\mathfrak{b}^{\mathfrak{a}}}{(\Delta n+\mathfrak{b})^{\mathfrak{a}}}) \quad (161)$$

□

Appendix D. Importance Sampling

Importance sampling is a technique that, instead of drawing samples r from the *nominal sampling distribution* p in order to estimate the expected value of some feature of the samples f, it draws the samples from an *importance distribution* q. These samples are then weighted according to the ratio between both distributions in order to obtain an estimate of $E[f]$ with a lower variance. This can vastly reduce the number of samples required to make statements with high confidence. It is well known (see Kahn and Marshall (1953)) that the optimal importance density satisfies

$$q(r) = \frac{f(r)p(r)}{E[f(r)]} \quad (162)$$

While calculating this exactly would require knowledge about the very value we want to estimate, it can often be approximated reasonably well. In our case, where each simulation run r depends on tens of thousands of random variables, we define a heuristic measure that roughly indicates how likely a run is to fail and partition the set of all possible runs into buckets using this measure. We then approximate the optimal q on the level of buckets.

As a first step, we now define the heuristic measure we use:

Definition 5. *For a deployment x, denote by*

$$i^x = E[L_n^x] + \sqrt{(1 - 0.01)/0.01 * Var[L_n^x]} \tag{163}$$

the upper bound of the 99% *confidence interval of a deployment's size in timestep n as given by Cantelli's inequality. For a given run r, pre-draw the parameters of all deployments that might arrive during the simulation run. Then consider the following extremely simplified simulation:*

1. *On the first of each month, 730 new deployments arrive.*
2. *Deployments only die at the end of the month after their maximum lifetime is reached. They do not die when they reach zero cores.*
3. *The cluster knows each deployment's exact type.*
4. *Deployment always take exactly their expected size.*
5. *Deployments are accepted into the cluster whenever $\sum_{\tilde{x} \in A_\pi(b)} i^x < 22000$.*

Denote by $X(n)$, $n \in [1, 36]$ the set of deployments in the cluster at the beginning of each month during this simplified simulation. Then the badness measure *BM of a run r is defined as*

$$BM(r) = max_n \sum_{\tilde{x} \in X(n)} i^x. \tag{164}$$

This is a reasonable (though highly heuristic) predictor of whether a run might produce a very large number failures. Most importantly, because it assumes away all randomness that occurs during the simulation run, it can be evaluated very quickly (< 1 second).

To properly utilize importance sampling, we now sort any simulation run r into one of three buckets based on their BM value: $I_1 = \{r : BM(r) \leq 25000\}$,
$I_2 = \{r : 25000 \leq BM(r) \leq 30000\}$, $I_3 = \{r : 30000 \leq BM(r)\}$. Before we can apply importance sampling, we calculate the probability for a given run to be in each of the buckets. For this, we calculate BM for $100,000$ runs. The resulting probabilities can be found in Table 4.

To sample runs from the different buckets with different weights, we employ a type of rejection sampling: Before starting a simulation run r, we evaluate $BM(r)$. Depending on the bucket I_i the run would result in, we then redraw with some probability $p_r(I_i)$ (i.e., all deployment parameters are discarded and redrawn) and block off all lower buckets (i.e., automatically rejecting any further redraws that would result in I_j, $j < i$). The highest bucket (in our case I_3) never gets redrawn, i.e., $p_r(I_3) = 0$. This scheme continues iteratively until we accept a run. This results in the following importance distribution.

Probabilities	I_1	I_2	I_3
$p(I_i)$	0.5699	0.4121	0.018
$p(I_i \mid \cap_{k\geq i} I_k)$	0.88319	0.9582	1
$p_r(I_i \mid \cap_{k\geq i} I_k)$	0.5369	0.8816	0

Table 4: Estimation of BM probabilities

Proposition 6. *For a run r with nominal probability $p(r)$ and BM such that $r \in I_i$, the above rejection scheme results in importance distribution q with*

$$q(r) = p(r|I_i)\frac{p(I_i \mid \cap_{k\geq i} I_k)(1 - p_r(I_i \mid \cap_{k\geq i} I_k))}{1 - p(I_i \mid \cap_{k\geq i} I_k)p_r(I_i \mid \cap_{k\geq i} I_k)}\Pi_{j<i}\frac{p(\cap_{k>j} I_k \mid \cap_{k\geq j} I_k)}{1 - p(I_j \mid \cap_{k\geq j} I_k)p_r(I_j \mid \cap_{k\geq j} I_k)} \tag{165}$$

Proof. We first show this for two buckets I_1 and I_2. With only two buckets, since I_2 is the highest bucket, it has an acceptance probability of 1.

It follows that for a run that would be in I_1, we redraw with probability $p_r(I_1)$ and otherwise accept. It thus holds

$$q(I_1) = p(I_1)((1 - p_r(I_1)) + p_r(I_1)q(I_1). \tag{166}$$

This is a geometric series and therefore

$$q(I_1) = \sum_{k=0}^{\infty} p(I_1)(1 - p_r(I_1))(p(I_1)p_r(I_1))^k \tag{167}$$

$$= \frac{p(I_1)(1 - p_r(I_1))}{1 - p(I_1)p_r(I_1)} \tag{168}$$

Similarly, it holds

$$q(I_2) = \sum_{i=0}^{\infty} p(I_2)(p(I_1)p_r(I_1))^i \tag{169}$$

$$= \frac{p(I_2)}{1 - p(I_1)p_r(I_1)} \tag{170}$$

Iteratively applying this argument to more than two buckets by dividing the top bucket into two buckets then yields

$$q(I_i) = \frac{p(I_i \mid \cap_{k\geq i} I_k)(1 - p_r(I_i \mid \cap_{k\geq i} I_k))}{1 - p(I_i \mid \cap_{k\geq i} I_k)p_r(I_i \mid \cap_{k\geq i} I_k)}\Pi_{j<i}\frac{p(\cap_{k>j} I_k \mid \cap_{k\geq j} I_k)}{1 - p(I_j \mid \cap_{k\geq j} I_k)p_r(I_j \mid \cap_{k\geq j} I_k)} \tag{171}$$

Finally, by Bayes' theorem it holds that $q(r) = q(r|I_i)q(I_i)$ and since it further holds that $q(r|I_i) = p(r|I_i)$, the statement of the proposition follows. □

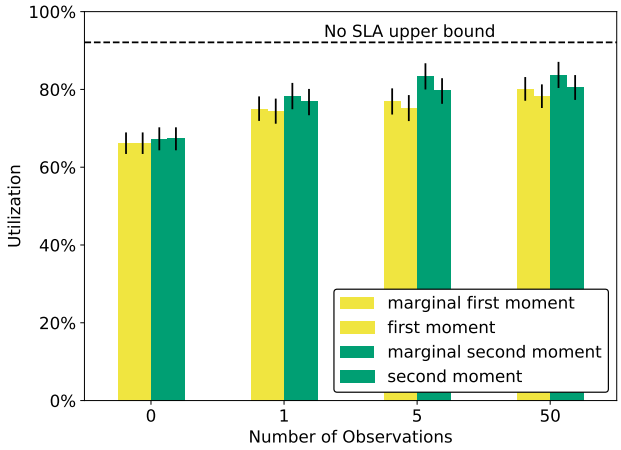

Figure 9: Ablation of policies with and without marginal heuristic (error bars indicate 95% bootstrap confidence intervals)

To find good rejection probabilities p_r that result in low sample variance, we did 500 runs for each bucket under the second moment policy with threshold 15000 to get a very rough estimate of f (i.e. the scale out failure probability) for each bucket. The rejection probabilities p_r are then calculated by combining Equation (162) and Equation (165). The resulting values can also be found in Table 4.

Appendix E. Ablation of Policies Without Marginal Heuristic

In this section, we ablate the simulation results for our policies with marginal heuristic with the same policies without these heuristic. Note again that without individual prior observations, no arriving deployment is ever marginal in the policy horizon. Thus, without prior observations, the marginal policies are equivalent to the policies without marginal heuristic.

We have simulated the policies without marginal heuristic for 1, 5, and 50 observations, with the same setup as in Section 5. In Figure 9, we contrast those results with the results for the marginal policies. We see that with just a few observations, the policies with and without marginal heuristic have a very similar performance (though the marginal heuristic still enables slightly higher utilization). This is not surprising, since relatively few arriving deployments are marginal at this level of prior information. Consequently, the utilization gap increases with more information. Both with 5 and 50 observations, the marginal second moment policy obtains a sizable utilization increase of more than 3% compared to the second moment policy without the heuristic.

4 The Competitive Effects of Variance-based Pricing

The content of this chapter has previously appeared in:

> Ludwig Dierks and Sven Seuken (2020) **The Competitive Effects of Variance-based Pricing**. Working Paper;
>
> Ludwig Dierks and Sven Seuken (2020) **The Competitive Effects of Variance-based Pricing**. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*.

The Competitive Effects of Variance-based Pricing

Ludwig Dierks and Sven Seuken

Department of Informatics, University of Zurich
{dierks,seuken}@ifi.uzh.ch

Abstract. In many markets, like electricity or cloud computing markets, providers incur large costs for keeping sufficient capacity in reserve to accommodate demand fluctuations of a mostly fixed user base. These costs are significantly affected by the unpredictability of the users' demand. Nevertheless, standard mechanisms charge fixed per-unit prices that do not depend on the variability of the users' demand. In this paper, we study a variance-based pricing rule in a two-provider market setting and perform a game-theoretic analysis of the resulting competitive effects. We show that an innovative provider who employs variance-based pricing can choose a pricing strategy that guarantees himself a higher profit than using fixed per-unit prices for any individually rational response of a provider playing a fixed pricing strategy. We then characterize all equilibria for the setting where both providers use variance-based pricing strategies. We show that, in equilibrium, the providers' profits may increase or decrease, depending on their cost functions. However, social welfare always weakly increases.

1 Introduction

In most markets with mostly fixed user bases, providers' costs are largely driven by how much buffer capacity they must keep in reserve. This, in turn, depends on the variance of their users' demand. However, the predominant pricing mechanisms employed in practice do not take this effect into account. Instead, prices are typically set on a per-unit basis, such that every user pays the same for the same product.

In electricity markets, for example, a provider has to make long-term supply decisions. But in real-time, supply and demand always have to be perfectly balanced, which requires a costly buffer infrastructure [Cramton, 2017]. If users would always consume (almost) the same amount of energy, this buffer could be far smaller than with widely varying user demands. Nevertheless, users pay simple per MW/h prices.

Next, consider the market for mobile data. Mobile network providers continuously expand their cell tower infrastructure to be able to satisfy their users' bandwidth needs under peak demand [López-Pérez *et al.*, 2009]. However, most end-users pay a fixed per-gigabyte-price, independent of when they consume it or how variable their demand is.[1]

Finally, consider cloud computing markets, where cloud providers must keep buffers of idle capacity in each compute cluster to handle changing resource demands of already running jobs. Handling the variance of cloud users is particularly difficult given

[1] Even in high-GDP countries, less than 10% of customers have unlimited data plans [Ericsson, 2017].

that the resource needs of an individual job may vary by a factor of 10 or 100 over time (see [Dierks *et al.*, 2019]). In this domain, the *mixture* of user types (i.e., their average demand variance) significantly affects the providers' need to supply buffer capacity. Nevertheless, most cloud resources are sold for a fixed price per core-hour, without regard to the variability of the users' demand.

Managing Demand via Sophisticated Pricing. Classic approaches for dealing with varying demand include dynamic pricing and congestion-based pricing [Muratori and Rizzoni, 2015, Rong *et al.*, 2018, Truong-Huu and Tham, 2014]. These approaches focus on flattening demand peaks. A big downside is that they make it unpredictable for users whether they can obtain the product at a given price when they need it. This puts providers who serve risk-averse users or users with relatively uniform but inelastic demand at a competitive disadvantage. In some markets, like cloud computing, this effect is so strong that providers never consider dynamic pricing for their *primary* market offerings [Dierks and Seuken, 2019]. The effect even greatly hinders the adoption of dynamic pricing in more suitable domains, like electricity markets [Joskow and Wolfram, 2012], where customers are instead often only exposed to fixed "time-of-use" tariffs [Urieli and Stone, 2016, Celebi and Fuller, 2012].

Variance-based Pricing. In this paper, we study *variance-based pricing*, where part of the price the users pay depends on the variance of their demand. Variance-based pricing was recently proposed by [Dierks *et al.*, 2019] to reduce costs by improving the demand prediction ability of a monopolistic cloud provider. For a provider, in general markets, variance-based pricing has the advantage that his low-variance users pay lower prices and are thus impacted less by the buffer requirements (which are mainly caused by high-variance users). This is not only fairer, but importantly, it incentivizes users to reduce their variance, which in turn reduces the provider's costs. In a monopolistic setting, the provider can obviously use variance-based pricing to increase his profits. However, in a competitive market environment, the effects are unclear, because the competitive pricing pressure by other providers may limit what he can achieve with variance-based pricing.

Overview of our Approach. We analyze a duopoly of providers who compete for a continuum of user types. We assume that the two providers consider the *long-term* effect of their pricing strategies on their profits. Since each provider has to provision enough capacity to guarantee service quality even during demand spikes, the cost per unit of a product depends on the average variance of the users he attracts. Each provider either conservatively employs constant per-unit prices or is willing to innovate and employ per-unit prices that linearly depend on each user's variance. We restrict ourself to linear prices here, because their simplicity makes them most plausible and marketable in practice. We show that, as long as a provider's costs are not far larger than the costs of his competitor, unilaterally switching to variance-based pricing can be used to obtain a higher profit for any reasonable constant response of the other provider. We then characterize all equilibria that arise if both providers employ variance-based pricing. We also show that, as long as providers are not symmetric in their cost functions, the profits of both providers often increase, as they can attract user types which they can better serve

due to their respective cost functions. Finally, we show that the welfare may decrease if only one provider employs variance-based pricing, but that it at least weakly increases if both employ variance-based pricing.

Related Work. Variance-based pricing is a type of price discrimination [Varian, 1989, Mussa and Rosen, 1978]. However, in contrast to the classic price discrimination settings [Moorthy, 1984, Blattberg and Wisniewski, 1989, Gallego *et al.*, 2006], variance-based pricing does not discriminate based on user preferences. Furthermore, in our model, there is neither a fixed marginal cost of supplying a given user nor do costs depend solely on the number of supplied products. As long as neither provider charges for variance, they cannot price discriminate at all and the problem becomes similar to a Bertrand competition [Baye and Kovenock, 2017]. In order to isolate the competitive effects caused by variance-based pricing, we assume that both providers offer the same product; thus, product differentiation (e.g., [Feng *et al.*, 2013]) does not take place.

2 Preliminaries

We consider a market setting with two providers and a continuum of users to which the providers want to sell their products. We study a simple two-period model: in the first period, the providers choose their pricing strategies; in the second period, the users choose a provider to buy the product from.

2.1 Formal Model

Each user is associated with a real-valued type $t \in [0, t_{max}]$. We keep this type general, but in practice it can be assumed to encode the variance of a user's demand. To keep the notation simple, we normalize each user's expected demand to 1. For a randomly chosen user, her type is distributed as a continuous random variable with pdf $f(t)$ and cdf $F(t)$. We do not model fluctuations in the mixture of user types over time.

Each provider's strategy space consists of his choice of price function ρ_i. We restrict $\rho_i = (p_i^f, p_i^\ell)$ to a *fixed* price p_i^f per unit of the product (independent of the user type) and a second *linearly* type-based charge $p_i^\ell t$ per unit of the product.[2] The overall payment per unit of the product for a user with type t is given by $\rho_i(t) = p_i^f + p_i^\ell t$. Going forward, we refer to providers that are willing to choose any $p_i^f, p_i^\ell \in \mathbb{R}^+$ as *innovative* and those who do not adopt the new pricing scheme and restrict themselves to $p_i^\ell = 0$ as *conservative*.

Given the provider's price functions, and depending on a user's own type, each user chooses to obtain the product either from provider 1 or 2. We denote a strategy profile for all user types by $\sigma(t) : [0, t_{max}] \to \{1, 2\}$. In this paper, we do not model

[2] As supply decisions (and therefore costs) do not depend on a user's true type but on the type the provider predicts for a user, we assume that providers know each realized user's type. In practice, user bases are often mostly fixed, such that a provider can learn each user's type over time.

the users' values for product consumption.[3] Consequently, the users' utility function is simply equal to their negative payments. Throughout the paper, given price functions (ρ_1, ρ_2), we assume that users only play *utility-maximizing* (i.e., payment minimizing) user strategy profiles. Formally, we assume that $\sigma(t) = argmin_i \{\rho_i(t)\}$ for all t.

As we will see, it is often in a provider's best interest to play essentially the same strategy as their opponent, which makes tie-breaking rules for the user sub-game very important. To avoid that tie breaking for $\rho_1 = \rho_2$ gives rise to arbitrary user strategy profiles that would not arise in practice, we restrict user strategy profiles to those at least one provider can enforce by deviating from (ρ_1, ρ_2) at only an infinitesimal profit loss. Formally, we say that a user strategy profile σ is *enforceable* at (ρ_1, ρ_2) if for one provider $i \in \{1, 2\}$ there exists a sequence of pricing functions $(\rho_i^k)_{k=1}^{\infty}$ such that σ is the limit of the user strategy profiles that are uniquely (up to a null set) utility maximizing for each (ρ_i^k, ρ_{-i}). With linearly type-based prices, this limits the user strategy profiles to any form where all users with type greater than some cutoff point $\hat{t}$ join one provider i and those with lower type join the other provider. To denote this, we also write $\sigma = [0, \hat{t}]_{\to i}$ or $\sigma = [\hat{t}, t_{max}]_{\to i}$. Note that this allows for the unique representation of any σ, as all other users choose the other provider. Further, we denote by $\mu(a, b)$ the average type of all users with type in $[a, b]$, i.e., $\mu(a, b) = \frac{\int_a^b f(t)tdt}{F(b)-F(a)}$.

A provider's costs for serving a given user do not only depend on how many units of his product he sells to that user, but also on how many units the provider has to supply for all users. Consequently, a provider's cost function $c_i(\sigma)$ is a function of the whole user strategy profile σ and cannot be separated into independent costs for individual users. We assume that $c_i(\sigma)$ is strictly increasing in the expected type of a randomly drawn user who joins provider i's market under σ. Overloading notation, we also write $c_i(a, b)$ for provider i's cost if all users in $[a, b]$ (and no other users) choose him.

In many applications, splitting a given population of users between two identical providers causes higher overall costs than if one provider would obtain all users, as that one provider could always provision for both sub-populations separately. We call such cost functions that are convex in relation to splitting the market *split-convex*, i.e. $c_i()$ is split-convex if for all $\hat{t} \in [0, t_{max}]$ it holds that

$$F(\hat{t})c_1(0, \hat{t}) + (1 - F(\hat{t}))c_1(\hat{t}, t_{\max}) \geq c_1(0, t_{max}).$$

If the inequality is strict for all $\hat{t} \in [0, t_{max}]$, we call the cost function *strictly split-convex*.

A provider's utility is his expected profit π_j *per customer in a population*, which is given by

$$\pi_j(\rho_1, \rho_2, \sigma) = \int_{t:\sigma(t)=j} (\rho_j(t) - c_j(\sigma))f(t)dt. \tag{1}$$

Given this model, we call a strategy ρ_i for provider $i \in \{1, 2\}$ an *individually rational response* to any given strategy ρ_{-i} of the other provider if there exists an enforceable utility-maximizing user strategy profile σ that gives provider i weakly positive profit, i.e., $\pi_i(\rho_i, \rho_{-i}, \sigma) \geq 0$.

[3] In the markets we study, essentially every user is served by some provider, as costs are very low compared to most users' values and competition ensures that prices are close to costs.

2.2 Equilibrium Concept

We primarily use the following equilibrium concept for our analysis.

Definition 1. *A tuple (ρ_1, ρ_2, σ) is a Bayes-Nash equilibrium (BNE) if σ is utility maximizing for (ρ_1, ρ_2) and, for $i \in \{1, 2\}$, there exist no $\hat{\rho}_i \neq \rho_i$ and $\hat{\sigma}$ such that $\hat{\sigma}$ is utility maximizing for $(\hat{\rho}_i, \rho_{-i})$ and*

$$\pi_i(\hat{\rho}_i, \rho_{-i}, \hat{\sigma}) > \pi_i(\rho_i, \rho_{-i}, \sigma). \tag{2}$$

Note that when $\rho_1 = \rho_2$, users are indifferent between all user strategy profiles, but our equilibrium definition mandates that tie breaking is done in such a way that no provider has an incentive to deviate infinitesimally only to secure himself a different user strategy profile. In practice, a combination of external factors and bounded rationality imply that providers do not move their prices to a tie or even infinitesimally close to each other, at best achieving $\epsilon-$BNEs. Essentially, price differentiations that are too small are not marketable.

Unfortunately, as we will see in Section 4.1, when one provider is innovative and the other is conservative, typically no Bayes-Nash-Equilibria exist. For any possible tuple (ρ_1, ρ_2, σ), at least one of the providers wants to deviate. To nonetheless make meaningful statements about these settings, note that an innovative provider typically first commits to his strategy by moving to variance-based prices and the other provider reacts to this innovation. We therefore introduce a second equilibrium concept, Stackelberg equilibria, that effectively views the game as a sequential move game: first one provider, the leader, commits to his strategy, anticipating the other providers reaction. The other provider, the follower, then reacts.

Because our model only contains 2 providers, after the leader has fixed his strategy, the follower does not face any additional competitive pressure in such an equilibrium concept. If the follower aims to maximize his profit, then this allows an innovative leader i to entice a conservative follower $-i$ to indirectly collude by setting a relatively low fixed-price p_i^f but a very high linear component p_i^ℓ. A conservative follower $-i$, being restricted to only setting one price for the whole market, then is best off also setting a high price. This would partly circumvent the competitive price pressure between providers and lead to relatively high profits at the users expense. As practical markets seldom contain exactly two providers, these profits would be overly optimistic. To show that the advantages of variance-based pricing do not depend on this artifact of the duopoly structure, we assume that the follower, instead of maximizing his profits, maximizes his market share as long as it provides him weakly positive profits. This can also be seen as implicitly modeling outside competitive pressure on the follower.

Definition 2. *A tuple (ρ_{-i}, σ) is a pessimistically competitive follower response to leader strategy ρ_i if σ is utility maximizing for (ρ_i, ρ_{-i}), results in non-negative profit, i.e., $\pi_i(\rho_i, \rho_{-i}, \sigma) \geq 0$, and there exists no $\hat{\rho}_{-i} \neq \rho_{-i}$ and $\hat{\sigma}$ such that $\hat{\sigma}$ is utility maximizing for $(\rho_i, \hat{\rho}_{-i})$ and*

$$\pi_i(\rho_i, \hat{\rho}_{-i}, \hat{\sigma}) \geq 0 \tag{3}$$

$$\int_{t:\sigma(t)=-i} f(t)dt > \int_{t:\sigma(t)=-i} f(t)dt. \tag{4}$$

A tuple (ρ_1, ρ_2, σ) is a pessimistically competitive Stackelberg equilibrium (pcSE) with leader i if (ρ_{-i}, σ) is a pessimistically competitive follower response to ρ_i and there exists no $\hat{\rho}_i$ with pessimistically competitive follower response $(\hat{\rho_{-i}}, \hat{\sigma})$ such that

$$\pi_i(\hat{\rho}_i, \hat{\rho}_{-i}, \hat{\sigma}) > \pi_i(\rho_i, \rho_{-i}, \sigma). \tag{5}$$

3 Profit Analysis with Conservative Providers

We first analyze the case where both providers are conservative, i.e., restricted to $p_1^\ell = p_2^\ell = 0$. Since they cannot split the market through pricing differences, the resulting game is similar to a classic Bertrand competition. Thus, if the providers' costs for the whole population are symmetric, they cannot extract any profit, while for non-symmetric providers, the provider with lower costs for serving the whole market can potentially extract the cost difference as a profit.

Proposition 1. *Let both providers be conservative, i.e., $p_1^\ell = p_2^\ell = 0$. W.l.o.g. assume $c_1(0, t_{max}) \leq c_2(0, t_{max})$. Then in any BNE (ρ_1, ρ_2, σ) the following holds:*

$$p_1^f = p_2^f \in [c_1(0, t_{max}), c_2(0, t_{max})] \tag{6}$$

and

$$\pi_1(\rho_1, \rho_2, \sigma) = p_1^f - c_1(0, t_{max}) \tag{7}$$
$$\pi_2(\rho_1, \rho_2, \sigma) = 0 \tag{8}$$

Proof. Note that any tuple (ρ_1, ρ_2, σ) with $p_1^f = p_2^f \in [c_1(0, t_{max}), c_2(0, t_{max})]$ and $\sigma = [0, t_{max}]_{\to 1}$ is a BNE as neither provider has an advantageous deviation. All users already choose provider 1, so decreasing his price only reduces his profit, while any price increase makes him lose all users. Since all users choose him, his profit is simply $\pi_1(\rho_1, \rho_2, \sigma) = p_1^f - c_1(0, t_{max})$. Any lower price for provider 2 on the other hand would lead to all users choosing him, but he would make a negative profit per user. Increasing his price would have no effect on his profit, as no users choose him. Without users, he trivially makes zero profit. As a special case, when $c_1(0, t_{max}) = c_2(0, t_{max})$, then, by the same argument, $p_1^f = p_2^f = c_2(0, t_{max})$ with $\sigma = [0, t_{max}]_{\to 2}$ is an additional BNE, with both providers obtaining zero profit.

We now show that these are the only BNEs, by showing that any other "potential BNE" leads to a contradiction. First, note that when $p_1^f < p_2^f$, every user strictly prefers provider 1. But there always exists a $\hat{p}_1^f$ with $p_1^f < \hat{p}_1^f < p_2^f$ for which every user still strictly prefers provider 1, but with a higher payment. Therefore, $p_1^f = p_2^f$ has to hold in equilibrium. If $p_1^f = p_2^f < c_1(0, t_{max})$, provider 1 would make a loss for every user. On the other hand, if $p_1^f = p_2^f > c_2(0, t_{max})$, then for any $\sigma = [0, t_{max}]_{\to i}$, the other provider $-i$ would have an advantageous deviation in any $\hat{p}_{-i}^f$ with $c_2(0, t_{max}) < \hat{p}_{-i}^f < p_1^f = p_2^f$. Taken together, this means that there can be no other BNE than those characterized by the proposition. □

Going forward, we denote BNEs with two conservative providers as *constant BNEs*. Note that, while all $p_1^f = p_2^f \in [c_1(0, t_{max}), c_2(0, t_{max})]$ are equilibrium prices, the only reason why any $p_1^f = p_2^f < c_2(0, t_{max})$ does not lead to a loss for provider 2 is because he obtains no users. This makes $p_1^f = c_2(0, t_{max})$ the only non-pathological BNE.

A provider publicly committing to a constant strategy also can not improve his profit further.

Proposition 2. *Let both providers be conservative, i.e., $p_1^\ell = p_2^\ell = 0$. Then no pessimistically competitive Stackelberg equilibrium can result in a higher profit for the leader than the non-pathological BNE.*

Proof. W.l.o.g. assume provider 1 is the leader and assume the tuple (ρ_1, ρ_2, σ) is a pessimistically competitive Stackelberg equilibrium with profit greater zero for provider 1. First note that for any p_1^f, p_2^f, setting $\hat{p}_1^f = \hat{p}_2^f = \min(p_1^f, p_2^f)$ without changing σ results in the same profit and market share for both providers. We can therefore w.l.o.g. assume $p_1^f = p_2^f$. If $p_1^f < c_2(0, t_{max})$, provider 1 makes less profit than in the non-pathological BNE. On the other hand, if $p_1^f > c_2(0, t_{max})$, then provider 2 would undercut his price and provider 1 would obtain zero users and no profit. Therefore, in any pessimistically competitive Stackelberg equilibrium provider 1 makes at most profit $c_2(0, t_{max}) - c_1(0, t_{max})$. □

4 Profit Analysis with Innovative Providers

In this section, we analyze the profit when one or both providers are innovative. As a first step, we show that, given a strategy of one provider, the other provider's profit is upper bounded by the profit he could achieve if he would play the same strategy as the first provider (and if he could select the utility-maximizing user strategy profile that is most favorable to him).

Proposition 3. *Assume provider i plays strategy $\rho_i = (p_i^f, p_i^\ell)$. Then, for all $\rho_{-i} \neq \rho_i$, $\rho'_{-i} = \rho_i$, and all σ that are utility maximizing for ρ_{-i}, ρ_i, it holds that $\pi_{-i}(\rho_{-i}, \rho_i, \sigma) = 0$ or $\pi_{-i}(\rho_{-i}, \rho_i, \sigma) < \pi_{-i}(\rho'_{-i}, \rho_i, \sigma)$.*

Proof. First note that for any $\rho_{-i} \neq \rho_i$, by the linearity of the price function, there exists a unique $\hat{t}$ at which payments in both markets are the same, i.e., $p_{-i}^f + \hat{t}p_{-i}^\ell = p_i^f + \hat{t}p_i^\ell$ (note that $\hat{t}$ can be outside $[0, t_{max}]$). For any utility-maximizing σ, users with type below $\hat{t}$ choose one provider and those above the other. Now consider some $\rho_{-i} \neq \rho_i$, $\pi_{-i}(\rho_{-i}, \rho_i, \sigma) > 0$ with $\hat{t}$ being the corresponding cutoff. If $p_{-i}^f \leq p_i^f$, $p_{-i}^\ell \geq p_i^\ell$, then any utility-maximizing user strategy profile is of the form $\sigma = [0, \bar{t}]_{\to -i}$

for $\bar{t} = min(\hat{t}, t_{max})$ and it holds that

$$\pi_{-i}(\rho_{-i}, \rho_i, \sigma) = \int_0^{\bar{t}} f(t)(p^f_{-i} + tp^\ell_{-i} - c_{-i}(0, \bar{t}))dt \tag{9}$$

$$= \int_0^{\bar{t}} f(t)(p^f_i + \hat{t}p^\ell_i - \hat{t}p^\ell_{-i} + tp^\ell_{-i} - c_{-i}(0, \bar{t}))dt \tag{10}$$

$$< \int_0^{\bar{t}} f(t)(p^f_i + \hat{t}p^\ell_i - c_{-i}(0, \bar{t}))dt \tag{11}$$

$$= \pi_{-i}(\rho_i, \rho_i, \sigma). \tag{12}$$

The proof works analogously for $p^f_{-i} \geq p^f_i$, $p^\ell_{-i} \leq p^\ell_i$. □

It directly follows that in all BNEs, both providers play the same strategy, even if one provider is conservative.

Corollary 1. *In all BNEs it holds that* $\rho_1 = \rho_2$.

We now separately consider the cases where either only one or both providers are willing to employ linear pricing.

4.1 One Provider is Innovative

For this section, assume that provider 1 is innovative, i.e., willing to employ variance-based pricing, and provider 2 is conservative, i.e., only willing to play $p^\ell_2 = 0$. Corollary 1 suggests that only one provider being innovative precludes the existence of a BNE. In the following theorem we show that this is indeed true, unless one providers' costs are "far lower" than the other providers.

Theorem 1. *Let provider* 2 *be conservative, i.e.,* $p^\ell_2 = 0$*, and provider* 1 *be innovative. Then there exists a BNE if and only if there exists no* $\hat{t} \in [0, t_{max}]$ *with*

$$c_1(0, t_{max}) - F(\hat{t})c_1(0, \hat{t}) \tag{13}$$

$$< (1 - F(\hat{t}))(c_1(0, t_{max}) - c_2(0, t_{max})). \tag{14}$$

If a BNE exists then it is equal to a constant BNE.

Proof. By Corollary 1, in all BNEs, both providers use constant pricing functions and for any constant pricing functions that are not part of a constant BNE there exist constant deviations. By Proposition 3, any deviation of provider 1 from a constant BNE is worse than him freely choosing the user strategy profile and not changing his pricing function. He can enforce any user strategy profile where he only obtains users with types lower than some $\hat{t}$. Thus, if $(\rho_1, \rho_2, [0, t_{max}]_{\to 1})$ is a constant BNE, then it is still a BNE when provider 1 is innovative as long as there exists no $\hat{t} < t_{max}$ such that $[0, \hat{t}]_{\to 1}$ leads to a higher profit than taking the whole market, i.e.,

$$c_1(0, t_{max}) - F(\hat{t})c_1(0, \hat{t}) \tag{15}$$

$$< (1 - F(\hat{t}))(c_1(0, t_{max}) - c_2(0, t_{max})). \tag{16}$$

Otherwise $[0, t_{max}]_{\to 1}$ cannot be a BNE user strategy profile.

Finally, for all $(\rho_1, \rho_2, [0, \hat{t}]_{\to 1})$ with $\hat{t} < t_{max}$, the users that choose provider 2 have a higher average variance than the average variance of all users. Consequently, $c_2(0, t_{max}) < c_2(\hat{t}, t_{max})$. As prices are constant, provider 2 either makes negative profit (and deviates) or he increases his profit by slightly decreasing his price and taking the whole market. □

Theorem 1 means that providers usually have an incentive to become innovative if their competition is conservative. However, becoming innovative and myopically optimizing profit can lead to non-existence of a BNE. In this case, an innovative provider should consider the additional market power he has due to his larger strategy space when choosing his strategy. Specifically, he should take a (myopically) non-optimal action to restrict his competitor's rational responses.

We now first show that, if provider 1 does so, he can guarantee himself strictly positive payoff whenever there is an interval of users for which his costs are lower than provider 2's costs for all users.

Theorem 2. *Let provider* 2 *be conservative, i.e.,* $p_2^\ell = 0$*. If there exists* $0 < \bar{t} < t_{max}$ *with*

$$c_1(0, \bar{t}) < c_2(0, t_{max}) \tag{17}$$

$$\textit{and } c_1(0, \bar{t}) < c_1(0, t_{max}) < 2c_1(0, \bar{t}), \tag{18}$$

then there exists a strategy ρ_1 *with* $p_1^\ell > 0$ *that guarantees provider* 1 *a non-negative payoff for any* p_2^f *and a strictly positive payoff for any individually rational response of provider* 2*.*

Proof. Note that for any $p_1^f < c_2(0, t_{max})$, $p_1^\ell > 0$, provider 2 can only achieve a positive profit if he plays $p_2^f = p_1^f + \hat{t}p_1^\ell$ for some $\hat{t} > 0$. To guarantee a positive payoff for provider 1 with such a p_1^f it therefore suffices that

$$\pi_1(\rho_1, \rho_2, \hat{t}) = \int_0^{\hat{t}} f(t)(p_1^f + tp_1^\ell - c_1(0, \hat{t}))dt > 0 \tag{19}$$

for all $\hat{t}$. For $0 < \bar{t} < t_{max}$ with $c_1(0, \bar{t}) < c_2(0, t_{max})$ and

$$c_1(0, \bar{t}) < c_1(0, t_{max}) < 2c_1(0, \bar{t}) \tag{20}$$

choose ρ_1 such that $p_1^f = c_1(0, \bar{t})$ and $p_1^\ell = \frac{c_1(0,\bar{t})}{\int_0^{\bar{t}} f(t)tdt}$. Then it holds for $\hat{t} \leq \bar{t}$:

$$\pi_1(\rho_1, \rho_2, \hat{t}) = \int_0^{\hat{t}} f(t)(p_1^f + tp_1^\ell - c_1(0, \hat{t}))dt \tag{21}$$

$$> \int_0^{\hat{t}} f(t)(p_1^f + tp_1^\ell - c_1(0, \bar{t}))dt \tag{22}$$

$$= \int_0^{\hat{t}} f(t)(tp_1^\ell)dt \tag{23}$$

$$> 0 \tag{24}$$

For $\hat{t} > \bar{t}$ it holds that

$$\pi_1(\rho_1, \rho_2, \hat{t}) \tag{25}$$

$$= \int_0^{\hat{t}} f(t)(p_1^f + tp_1^\ell - c_1(0, \hat{t}))dt \tag{26}$$

$$= \int_0^{\hat{t}} f(t)(t\frac{c_1(0,\bar{t})}{\int_0^{\bar{t}} f(t)tdt} - c_1(0, \hat{t}) - c_1(0, \bar{t}))dt \tag{27}$$

$$> \int_0^{\hat{t}} f(t)(c_1(0, \bar{t}) - c_1(0, \hat{t}) - c_1(0, \bar{t}))dt \tag{28}$$

$$= \int_0^{\hat{t}} f(t)(2c_1(0, \bar{t}) - c_1(0, \hat{t})dt \tag{29}$$

$$>0 \tag{30}$$

Therefore, ρ_1 guarantees provider 1 positive profit for $\sigma = [0, \hat{t}]_{\to 1}$ for any $\hat{t} \in (0, t_{max}]$. □

In words, Theorem 2 says that provider 1 always has a strategy which is individually rational (independent of the other provider's action) and that results in strictly positive profit for all individually rational responses of provider 2. This is especially attractive if provider 1 could only obtain zero profit as a conservative provider. As we can see from the proof, a possible such pricing strategy is given by $p_1^f = c_1(0, \bar{t})$ and $p_1^\ell = \frac{c_1(0,\bar{t})}{\int_0^{\bar{t}} f(t)tdt}$. While this strategy does neither constitute a BNE nor a pessimistically competitive Stackelberg equilibrium (i.e., the leader might be able to do even better), it gives us a lower bound for the profit achievable by an innovative leader in pessimistically competitive Stackelberg equilibria.

This still leaves the question whether a provider who obtains positive profit as a conservative provider should also become innovative. As the following proposition shows, the answer is usually yes.

Theorem 3. *Let provider 2 be conservative, i.e., $p_2^\ell = 0$. If provider 1 obtains strictly positive profit in some constant BNE then there exists a strategy ρ_1 with $p_1^\ell > 0$ that, for any individually rational response of provider 2, guarantees provider 1 greater profit than in any constant BNE.*

Proof. Recall that by Proposition 1, if provider 1 has positive profit in some constant equilibrium then

$$c_1(0, t_{max}) < c_2(0, t_{max}). \tag{31}$$

Let $\bar{t}$ be a type such that

$$\bar{t} = argmin_t \frac{c_2(t, t_{max}) - c_2(0, t_{max})}{t}. \tag{32}$$

For any $\epsilon > 0$ with $\epsilon < \mu(0, t_{max})\frac{c_2(\bar{t},t_{max})-c_2(0,t_{max}))}{\bar{t}}$, let $\rho_1 = (p_1^f, p_1^\ell)$ with $p_1^f = c_2(0, t_{max}) - \epsilon$ and $p_1^\ell = \frac{c_2(\bar{t},t_{max})-c_2(0,t_{max})}{\bar{t}}$. Then for any p_2^f provider 2 obtains no

users or there exists a $t > \hat{t}$ such that he obtains all users with type $t > \hat{t}$ and his profit is given by

$$\pi_2(\rho_1, \rho_2, \sigma) = \int_{\hat{t}}^{t_{max}} f(t)(p_2^f - c_2(\hat{t}, t_{max}))dt \tag{33}$$

$$= \int_{\hat{t}}^{t_{max}} f(t)(p_1^f + \hat{t}p_1^\ell - c_2(\hat{t}, t_{max}))dt \tag{34}$$

$$\leq \int_{\hat{t}}^{t_{max}} f(t)(p_1^f + \hat{t}\frac{c_2(\hat{t}, t_{max}) - p_1^f + \epsilon}{\hat{t}} \tag{35}$$

$$- c_2(\hat{t}, t_{max}))dt \tag{36}$$

$$= 0 - (F(t_{max}) - F(\hat{t}))\epsilon \tag{37}$$

Therefore, provider 2 has no individually rational response for which he obtains any users. It follows that for any individually rational response of provider 2 and any $\epsilon < \mu(0, t_{max})\frac{c_2(\bar{t}, t_{max}) - c_2(0, t_{max})}{\bar{t}}$ provider 1's profit is

$$\pi_1(\rho_1, \rho_2, \sigma) \tag{38}$$

$$= \int_0^{t_{max}} f(t)(p_1^f + tp_1^\ell - c_1(0, t_{max}))dt \tag{39}$$

$$= c_2(0, t_{max}) - c_1(0, t_{max}) - \epsilon \tag{40}$$

$$+ \mu(0, t_{max})\frac{c_2(\bar{t}, t_{max})) - c_2(0, t_{max})}{\bar{t}} \tag{41}$$

$$> c_2(0, \hat{t}) - c_1(0, \hat{t}) \tag{42}$$

By Proposition 1, the profit therefore is greater than the profit in any constant BNE. □

Note that, in contrast to the strategies described by the proof of Theorem 2, the strategies described by the proof of Theorem 3 are not guaranteed to be individually rational responses to *every* possible action provider 2 might play. However, they *are* guaranteed to be individually rational responses to every constant BNE strategy of provider 2 and again constitute a lower bound for the profit achievable by an innovative leader in pessimistically competitive Stackelberg equilibria.

Taken together, it also follows from these two results that, under fairly mild conditions, innovating to variance-based prices in the face of conservative competition will increase a providers profit in any pessimistically competitive Stackelberg equilibrium.

Theorem 4. *Let provider* 2 *be conservative, i.e.,* $p_2^\ell = 0$*. If provider* 1 *obtains strictly positive profit in some constant BNE or there exists* $0 < \bar{t} < t_{max}$ *with*

$$c_1(0, \bar{t}) < c_2(0, t_{max}) \tag{43}$$

$$\textit{and } \; c_1(0, \bar{t}) < c_1(0, t_{max}) < 2c_1(0, \bar{t}), \tag{44}$$

then in any pessimistically competitive Stackelberg equilibrium (ρ_1, ρ_2, σ) *with leader* 1 *it holds that provider* 1 *is innovative, i.e., plays* $p_1^\ell > 0$*, and obtains higher profit than in any constant BNE.*

Proof. From combining Theorem 2 and Theorem 3, it follows that the provider can obtain higher profit than in any constant BNE by playing $p_1^\ell > 0$. On the other hand, Proposition 2 gives us that no strategy with $p_1^\ell = 0$ can result in a profit that is higher than the best constant BNE. □

While the strategies presented by Theorem 2 and Theorem 3 are fully adequate to achieve higher profit than any constant strategy could, they are typically not optimal and do not result in pessimistically competitive Stackelberg equilibrium strategies. Finding Stackelberg equilibria is generally hard, as they are defined as the solutions to the following (potentially non-differentiable and non-convex) bilevel optimization problem.

$$\max_{p_1^f \in \mathbb{R}^+, p_1^f \in \mathbb{R}^+} \quad \pi_1((p_1^f, p_1^\ell), (p_2^f, 0), [0,t]_{\to 1}) \tag{45a}$$

$$\text{subject to} \quad \max_{p_2^f \in [p_1^f, p_1^f + p_1^\ell t_{max}]} \quad \int_t^{t_{max}} f(t)dt \tag{45b}$$

$$\text{subject to} \quad \pi_2((p_1^f, p_1^\ell), (p_2^f, 0), [0,t]_{\to 1}) \geq 0 \tag{45c}$$

$$t = \frac{p_2^f - p_1^f}{p_1^\ell}. \tag{45d}$$

How to best solve this depends on the specific forms the cost functions and the distribution over user types take. As the problem only contains three variables, evolutionary algorithms such as differential evolution can typically be employed to find near-optimal solutions relatively quickly, though they lack optimality guarantees. A good overview of potential solution methods can be found in [Sinha *et al.*, 2017].

4.2 Both Providers are Innovative

Once one provider starts to employ linear pricing, the other provider might at some point also want to follow. Consequently, we now look at the case where both providers are innovative. When both providers employ linear pricing, the first provider loses much of the additional power he had when the other provider stayed conservative. Consequently, there is no general guarantee that he can still improve his profit. But as long as the cost functions are strictly split-convex, he can still choose a strategy which guarantees him that profits do not decrease compared to any constant BNE (for any individually rational response of the other provider).

Proposition 4. *Assume the cost function of provider* 2 *is strictly split-convex. Then there exists a strategy* ρ_1 *with* $p_1^\ell > 0$ *that, for any individually rational response of provider* 2*, guarantees provider* 1 *greater or equal profit than in any constant BNE.*

Proof. If provider 1 obtains 0 profit in all constant BNEs, nothing has to be shown. Otherwise, assume provider 1 plays $\rho_1 = (p_1^f, p_1^\ell)$ with $p_1^\ell = \frac{d}{dt}|_{t_{max}} c_2(0,t)$ and $p_1^f +$

$p_1^\ell \mu(0, t_{max}) = c_2(0, t_{max})$. By Proposition 3, we know that for all ρ_2 with utility-maximizing $\sigma = [0, \hat{t}]_{\to 2}$ it holds that $\pi_2(\rho_1, \rho_2, \sigma) \leq \pi_2(\rho_1, \rho_1, \sigma)$ and from strict split-convexity it follows that

$$
\begin{aligned}
& c_2(0, \hat{t}) && (46)\\
>& c_2(0, t_{max}) && (47)\\
& + \frac{d}{dt}|_{t_{max}} c_2(0, t)(\mu(0, \hat{t}) - \mu(0, t_{max})) && (48)\\
=& p_1^f + p_1^\ell \mu(0, t_{max}) + p_1^\ell(\mu(0, \hat{t}) - \mu(0, t_{max})) && (49)\\
=& p_1^f + p_1^\ell \mu(0, \hat{t}) && (50)
\end{aligned}
$$

and therefore

$$
\begin{aligned}
& \pi_2(\rho_1, \rho_1, \sigma) = \int_0^{\hat{t}} f(t)(p_1^f + t p_1^\ell - c_2(0, \hat{t}))dt && (51)\\
<& \int_0^{\hat{t}} f(t)(p_1^f + t p_1^\ell - p_1^f - p_1^\ell \mu(0, \hat{t}))dt && (52)\\
=& \int_0^{\hat{t}} f(t)(t p_1^\ell - p_1^\ell \mu(0, \hat{t}))dt && (53)\\
=& 0. && (54)
\end{aligned}
$$

The proof works analogously for $\sigma = [0, \hat{t}]_{\to 1}$. Therefore, any individually rational response of provider 2 results in user strategy profile $\sigma = [0, t_{max}]_{\to 1}$. For this strategy profile, provider 1 has profit $\pi_1(\rho_1, \rho_2, \sigma) = c_1(0, t_{max}) - c_2(0, t_{max})$, which by Proposition 1 is the highest possible profit for any constant BNE. □

Similarly to what we have seen with a conservative follower, this generic strategy lower bounds the profit in any pessimistically competitive Stackelberg equilibrium with a potentially innovative follower.

Theorem 5. *If the cost function of provider 2 is strictly split-convex, then any pessimistically competitive Stackelberg equilibrium (ρ_1, ρ_2, σ) results in (weakly) higher profit than any constant BNE.*

Proof. Follows directly from Proposition 4. □

Strictly higher profit can not be guaranteed. For example, if both providers are symmetric and their costs are split-convex, moving to linear prices still cannot lead to any profit in equilibrium.

Proposition 5. *Assume providers are symmetric, i.e., $c_1(\cdot) = c_2(\cdot)$, and costs are split-convex. Then there can be no BNE with strictly positive profit for either provider and no pessimistically competitive Stackelberg equilibrium with strictly positive profit for the leader.*

Proof. If, w.l.o.g., all users prefer provider 1 and he obtains strictly positive payoff, provider 2 obtains zero profit but can deviate to $p_2^f = p_1^f - \epsilon, p_2^\ell = p_1^\ell$ to obtain strictly positive payoff for $\epsilon > 0$ small enough. Therefore, in any potential BNE, the market is split between providers or both obtain zero profit. Assume (ρ_1, ρ_2, σ) is a BNE and w.l.o.g. $\sigma = [0, \hat{t}]_{\to 1}$ for some $0 < \hat{t} < t_{max}$. By Corollary 1 we can assume $\rho_1 = \rho_2$. Then, for any $\epsilon > 0$ and $\hat{\rho}_2 = (p_2^f - \epsilon, p_2^\ell)$ all users prefer provider 2, resulting in profit

$$\pi_2(\rho_1, \hat{\rho}_2, [0, t_{max}]_{\to 2}) \tag{55}$$

$$= \int_0^{t_{max}} f(t)(p_2^f - \epsilon + t p_2^\ell - c_2(0, t_{max}))dt \tag{56}$$

$$= \int_0^{t_{max}} f(t)(p_2^f + t p_2^\ell)dt - \epsilon - c_2(0, t_{max}) \tag{57}$$

$$\geq \int_0^{t_{max}} f(t)(p_2^f + t p_2^\ell)dt - \epsilon \tag{58}$$

$$- (1 - F(\hat{t}))c_2(\hat{t}, t_{max}) + F(\hat{t})c_2(0, \hat{t}) \tag{59}$$

$$= \pi_2(\rho_1, \rho_2, \sigma) + \pi_1(\rho_1, \rho_2, \sigma) - \epsilon \tag{60}$$

If $\pi_2(\rho_1, \rho_2, \sigma) > 0$ or $\pi_1(\rho_1, \rho_2, \sigma) > 0$, then it follows that for ϵ small enough, $\pi_2(\rho_1, \hat{\rho}_2, \hat{\sigma}) > \pi_2(\rho_1, \rho_2, \sigma)$, contradicting our assumption that (ρ_1, ρ_2, σ) is a BNE. Thus, it must hold that $\pi_2(\rho_1, \rho_2, \sigma) = 0$ and $\pi_1(\rho_1, \rho_2, \sigma) = 0$. The equivalent arguments also holds for the leader in a pessimistically competitive Stackelberg equilibrium, as the follower will always take the whole market if he can do so without obtaining negative profit. □

When cost functions are not split-convex, symmetric providers still always obtain the same profit in any BNE, even though it can be positive.

Proposition 6. *Assume providers are symmetric, i.e., $c_1(\cdot) = c_2(\cdot)$. Then, in all BNEs it holds that $\pi_1(\rho_1, \rho_2, \sigma) = \pi_2(\rho_1, \rho_2, \sigma)$.*

Proof. To see this, note that for symmetric providers, whoever has lower profits could switch users with the other one by making an infinitesimal change to his price function. □

By definition, whether a tuple (ρ_1, ρ_2, σ) is a BNE is decided via a three-dimensional condition space, as the profit has to be better than the profit for any other tuple $(\hat{\rho_1}, \hat{\rho_2}, \hat{\sigma})$. This makes it very hard to evaluate whether a given tuple is a BNE. The following theorem instead characterizes equilibria by a one-dimensional condition space, greatly reducing the complexity of checking candidate equilibria.

Theorem 6. *A tuple $(\rho_1, \rho_2, [0, \hat{t}]_{\to i})$ is a BNE if and only if $\rho_1 = \rho_2$ and*

$$F(\hat{t})c_i(0, \hat{t}) - F(a)c_i(0, a) \tag{61}$$

$$\leq (F(\hat{t}) - F(a)(p_i^f + \mu(a, t)p_i^\ell) \tag{62}$$

$$\leq (1 - F(a))c_{-i}(a, t_{max}) - (1 - F(\hat{t}))c_{-i}(\hat{t}, t_{max}) \tag{63}$$

for all $0 \leq a \leq t_{max}$*. If the providers are not symmetric, it also has to hold for all* $0 \leq a \leq t_{max}$ *that*

$$F(a)(p_i^f + \mu(0,a)p_i^\ell - c_{-i}(0,a)) \tag{64}$$
$$\leq (1 - F(\hat{t}))(p_i^f + \mu(\hat{t}, t_{max})p_i^\ell - c_{-i}(\hat{t}, t_{max})) \tag{65}$$

and

$$(1 - F(a))(p_i^f + \mu(a, t_{max})p_i^\ell - c_i(a, t_{max})) \tag{66}$$
$$\leq F(\hat{t})(p_i^f + \mu(0,\hat{t})p_i^\ell - c_i(0,\hat{t})). \tag{67}$$

The proof is provided in Appendix A.

While Theorem 6 fully characterizes all BNEs, it is a very technical characterization. It can be used to *check* whether a given tuple (ρ_1, ρ_2, σ) *is* a BNE, but it does not enable an easy *search procedure* for finding candidate BNEs. This is particularly important because a $(\rho_1, \rho_2, [0,\hat{t}]_{\to i})$ that satisfies the conditions of Theorem 6 does not always exist. Thus, even if both providers are innovative, there are cases where no BNE exists. The following corollary addresses this issue, identifying a small subset of user strategy profiles that can be part of a BNE and reducing the search for corresponding provider strategies to a one-dimensional search.

Corollary 2. *If a tuple* (ρ_1, ρ_2, σ) *with* $0 < \hat{t} < t_{max}$ *is a BNE and the cost functions are differentiable, then it holds*

$$\frac{d}{dt}|_{\hat{t}} F(t)c_2(0,t) \tag{68}$$
$$= f(\hat{t})(p^f + \hat{t}p^\ell). \tag{69}$$
$$= \frac{d}{dt}|_{\hat{t}} - (1 - F(t))c_1(t, t_{max}) \tag{70}$$

Proof. Note that for $t = \hat{t}$ it trivially holds

$$F(\hat{t})c_2(0,\hat{t}) - F(t)c_2(0,t) \tag{71}$$
$$= (F(\hat{t})) - F(t)(p_1^f + \mu(t,t)p_1^\ell) \tag{72}$$
$$= (1 - F(t))c_1(t, t_{max}) - (1 - F(\hat{t}))c_1(\hat{t}, t_{max}). \tag{73}$$

Theorem 6 further gives us that the three expressions have the same ordering for all t, especially for all t in any small neighborhood around $\hat{t}$. Therefore, all three expressions need to have the same derivative in t at point $t = \hat{t}$. □

Given a cutoff point $\hat{t}$, all potential BNE price functions lie on a line defined by Equation (69). To find a BNE, all that remains to be done is to check whether $\hat{t}$ with any of the (p^f, p^ℓ) on that line satisfy the conditions of Theorem 6.

5 Welfare Analysis

In this section, we analyze the impact of variance-based pricing on social welfare. Since all users always fulfill their demand, the social welfare is simply the negative sum of the expected costs of both providers, i.e., $w(\rho_1, \rho_2, [0,t]_{\to i}) = -F(\hat{t})c_i(0,\hat{t}) - (1 - F(\hat{t}))c_{-i}(\hat{t}, t_{max})$. The social welfare in any constant BNE then follows directly from Proposition 1.

Corollary 3. *Let both providers be conservative and w.l.o.g. assume* $c_1(0,t_{max}) \leq c_2(0,t_{max})$. *Then the social welfare in any BNE is* $w(\rho_1,\rho_2,\sigma) = -c_1(0,t_{max})$.

Proof. Follows directly from Proposition 1. □

If only one provider is innovative, then whether the social welfare increases or decreases depends on the relative cost functions of the two providers. This is because the innovative provider typically employs his additional market power to force the conservative provider to give up the low variance part of the market, even if the conservative provider would be better suited to serve those users. But if both are innovative, he loses this power and in BNE, the social welfare cannot decrease compared to any constant BNE.

Proposition 7. *W.l.o.g. assume* $c_1(0,t_{max}) \leq c_2(0,t_{max})$. *When both providers are innovative, then the social welfare in any BNE* (ρ_1,ρ_2,σ) *is higher than the social welfare in any constant BNE, i.e.,* $w(\rho_1,\rho_2,\sigma) \geq -c_1(0,t_{max})$.

Proof. Recall that the social welfare is $w(\rho_1,\rho_2.[0,t]_{\to i}) = (-1)\left(F(\hat{t})c_i(0,\hat{t}) + (1-F(\hat{t}))c_{-i}(\hat{t},t_{max})\right)$. Thus, whenever all users choose provider 1, the social welfare is the same as in any constant BNE, i.e., $w(\rho_1,\rho_2,[0,t_{max}]_{\to 1}) = -c_1(0,t_{max})$.

We now show the claim of the proposition by contradiction. Assume $(\rho_1,\rho_2,[0,\hat{t}]_{\to i})$ for $i \in 1,2$ is an innovative BNE where the social welfare is strictly lower than the social welfare in any constant BNE. Then the social welfare under $(\rho_1,\rho_2,[0,\hat{t}]_{\to i})$ is also strictly lower than under $(\rho_1,\rho_2,[0,t_{max}]_{\to 1})$, i.e., the sum of the expected costs is strictly higher; formally: $F(\hat{t})c_i(0,\hat{t})+(1-F(\hat{t}))c_{-i}(\hat{t},t_{max}) > c_1(0,t_{max})$. Additionally, by Corollary 1, we can assume that $\rho_1 = \rho_2$. This means that the payment of any user is independent of which provider he chooses, and therefore the sum of the revenues of both providers does not change between $(\rho_1,\rho_2,[0,\hat{t}]_{\to i})$ and $(\rho_1,\rho_2,[0,\hat{t}]_{\to 1})$. Taken together, when going from the first profile to the second profile, the sum of the revenues stay the same but the costs strictly decrease, which implies that $\pi_1(\rho_1,\rho_2,[0,t_{max}]_{\to 1}) > \pi_1(\rho_1,\rho_2,[0,\hat{t}]_{\to i})+\pi_2(\rho_1,\rho_2,[0,\hat{t}]_{\to i})$. Therefore, $(\rho_1,\rho_2,[0,\hat{t}]_{\to i})$ can not be a BNE, a contradiction. □

6 Numerical Example

In this section, we illustrate our results via a simple numerical example. We assume that user types are uniformly distributed on $[0,1]$ and that the users' value is $v = 2$. Further, we assume provider 1 has cost function $c_1(a,b) = 0.0125 + \mu(a,b)^2$ and provider 2 has cost function $c_2(a,b) = 0.2 + \frac{\mu(a,b)^2}{4}$, where $\mu(a,b)$ denotes the average type of

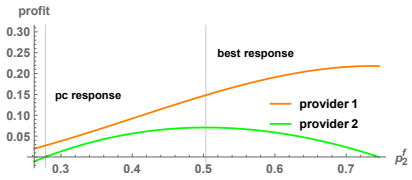

Fig. 1: Profit of both providers for all conservative responses of provider 2 to provider 1 playing $(p_1^f = 0.215, p_1^\ell = 0.5309)$ following Theorem 2.

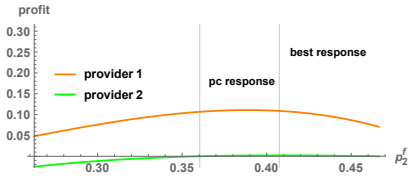

Fig. 2: Profit of both providers for all conservative responses of provider 2 to provider 1 playing the (approximate) pessimistically competitive Stackelberg strategy $(p_1^f = 0.2001, p_1^\ell = 0.2662)$.

all users in $[a, b]$ (as previously defined in Section 2.1). Thus, provider 1 has a lower cost for low types but a higher cost for high types than provider 2, and both providers have the same cost for the whole user population. From Proposition 1, we know that when both providers are conservative, there are only zero-profit BNEs. They occur at $p_1^f = p_2^f = 0.2625$ with a welfare of -0.2625.

We now consider each provider unilaterally switching to linear prices. First, if provider 1 innovates using a generic strategy as described in Theorem 2 (e.g., $\rho_1 = (p_1^f = 0.215, p_1^\ell = 0.5309)$) then Figure 1 shows that he can obtain a sizable profit increase of 0.1482 whenever provider 2 plays a profit-optimal best response. Social welfare in this case improves to -0.2062, as each provider obtains a part of the market he can more efficiently serve than his competitor. If provider 2 instead faces additional outside competitive pressure or for some other reason plays a pessimistically competitive response (marked "pc response" in the figure), then provider 1 at least still obtains a profit of 0.027, while social welfare only improves to -0.2470. If provider 1 instead already expects a pessimistically competitive response and aims to achieve a pessimistically competitive Stackelberg equilibrium, then he can further increase his worst case profit. Applying a differential evolution search to the optimization formulation (45) yields $(p_1^f = 0.200114, p_1^\ell = 0.266251)$ as an approximate equilibrium strategy for the innovating provider 1. As can be seen in Figure 2, this results in an pessimistically competitive equilibrium profit of 0.1067. Unfortunately, if the follower unexpectedly does play a profit-optimizing best response, then provider 1's profit is still only 0.1087.

Figure 3 and Figure 4 show the analogous results when provider 2 becomes innovative instead. Contrasting Figure 3 with Figure 1, we see that, with the generic strategies from Theorem 2, provider 1 innovating leads to the overall better result for both providers. This is not surprising, considering that, with these strategies, the innovative provider obtains the lower type portion of the market in which provider 1 has lower

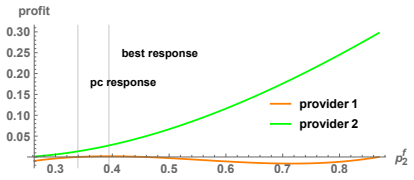

Fig. 3: Profit of both providers for all conservative responses of provider 1 to provider 2 playing $(p_2^f = 0.2506, p_2^\ell = 0.6188)$ following Theorem 2.

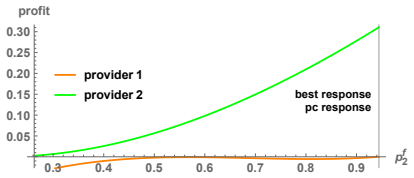

Fig. 4: Profit of both providers for all conservative responses of provider 1 to provider 2 playing the (approximate) pessimistically competitive Stackelberg strategy $(p_2^f = 0.2019, p_2^\ell = 0.7424)$.

costs. Nonetheless, if provider 2 innovates, he still obtains a profit of 0.0276 at provider 1's best constant response of $p_1^f = 0.3941$ and a profit of 0.0135 at the pessimistically competitive response of $p_1^f = 0.3394$. Since provider 2 uses the power of his larger strategy space as an innovative provider to force provider 1 to obtain a high type population interval (for which provider 1 has higher costs) social welfare unsurprisingly decreases to -0.3846. This generic strategy is far from optimal for provider 2 though. If provider 2 instead aims for a pessimistically competitive equilibrium and plays $(p_2^f = 0.2019, p_2^\ell = 0.7324)$, then we see in Figure 4 that he can force provider 1 completely out of the market. This leads to him obtaining a profit of 0.3106 and restoring welfare back to -0.2625.

For the case where both providers are willing to employ linear pricing, Corollary 2 provides us with conditions on candidate equilibrium user strategy profiles. For our example, we can use those conditions to find four cutoff points: $\sigma = [0, 0.595]_{\to 1}$, $\sigma = [0.5431, 1]_{\to 1}$, $\sigma = [0, 1]_{\to 1}$ and $\sigma = [0, 0]_{\to 1}$. All of these except for $\sigma = [0, 0.595]_{\to 1}$ do not satisfy Theorem 6 and are eliminated. For $\sigma = [0, 0.595]_{\to 1}$ any $p_1^f = p_2^f \in [0, 0.0409]$, $p_1^\ell = p_2^\ell = \frac{0.2784 - p_1^f}{0.595}$ satisfy Theorem 6 and form equilibrium pricing strategies. To visualize an example BNE, Figure 5 shows the profit of both providers for $p_1^f = p_2^f = 0$ and $p_1^\ell = p_2^\ell = 0.4676$ for different utility-maximizing $\sigma = [0, t]_{\to 1}$. We see that neither provider wants to deviate to enforce a different user strategy profile $\sigma = [0, t]_{\to 1}$ than $\sigma = [0, 0.595]_{\to 1}$. Social welfare at this BNE is -0.2055, which is even slightly better than when only provider 1 was innovative. However, the increased competition leads to a significantly lower profit for both providers than when only provider 1 was innovative. This suggests that, as long as there is not too much outside competitive pressure, a conservative provider who is forward-looking

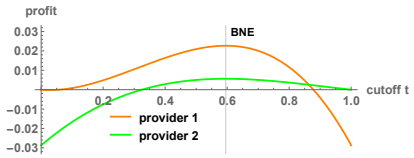

Fig. 5: Profit of both providers for different $\sigma = [0, t]_{\to 1}$

and properly anticipates the possible BNEs may prefer to stay conservative if the other provider is already innovative.

7 Conclusion

In this paper, we have studied the competitive effects of providers utilizing linear variance-based (or type-based) pricing rules in settings where a provider's costs depend on the average type of all his users. We have shown that, while a single provider innovating often leads to non-existence of BNEs, the innovative provider can exert the additional market power he has due to his larger strategy space to unilaterally set prices that increase his profit for all individually rational responses of a conservative provider. We have further characterized all equilibria where both providers employ linear pricing. While much of the additional market power of an innovative provider is lost once the other provider also adopts linear pricing, the increased strategy space allows providers to split the market more closely along differences in their cost functions. This often increases both providers' profits and social welfare. While our general insights translate to settings with more than two providers, the strategic considerations and the BNEs quickly become too complex to provide useful insights. In future work, it would be interesting to study secondary effects like incentivizing users to actively lower their variance. Furthermore, to deploy variance-based pricing in practice, it would be important to develop good machine learning algorithms to effectively learn the users' types over time.

A Proof of Theorem 6

Proof. A tuple $(\rho_1, \rho_2, [0, \hat{t}]_{\to i})$ is a BNE if $[0, \hat{t}]_{\to i}$ is utility maximizing and no provider has a profitable deviation. From Proposition 3 we know that any deviation with different prices is worse than keeping the same prices and choosing the deviating provider's most-preferred utility-maximizing user strategy profile. Thus, the tuple is a BNE if and only if moving to any different user strategy profile (weakly) decreases both providers' profits. W.l.o.g. assume $i = 2$, and thus, in the candidate BNE we consider, provider 2 obtains the low-variance users. We first consider how profits change under different user strategy profiles for which provider 2 still obtains the low-variance users. Changing to

any $[0,a]_{\to 2}$ with $0 \leq a < \hat{t}$ yields a profit change for Provider 1 of

$$(1-F(\hat{t}))(c_1(\hat{t},t_{max}) - c_1(a,t_{max}) \tag{74}$$

$$+ \int_a^{\hat{t}} f(t)(p_1^f + tp_1^\ell - c_1(a,t_{max}))dt \tag{75}$$

$$=(1-F(\hat{t}))c_1(\hat{t},t_{max}) - (1-F(a))c_1(a,t_{max}) \tag{76}$$

$$+ (F(\hat{t}) - F(a))(p_1^f + \mu(a,t)p_1^\ell) \tag{77}$$

while with $\hat{t} < a < t_{max}$ it yields a change of

$$(1-F(a))(c_1(\hat{t},t_{max}) - c_1(a,t_{max}) \tag{78}$$

$$- \int_{\hat{t}}^{a} f(t)(p_1^f + tp_1^\ell - c_1(\hat{t},t_{max}))dt \tag{79}$$

$$=(1-F(\hat{t}))c_1(\hat{t},t_{max}) \tag{80}$$

$$- (1-F(a))c_1(a,t_{max}) \tag{81}$$

$$+ (F(\hat{t}) - F(a))(p_1^f + \mu(a,t)p_1^\ell). \tag{82}$$

Similarly, the profit change for provider 2 for $a < \hat{t}$ is

$$F(a)(c_2(0,\hat{t}) - c_2(0,a)) \tag{83}$$

$$- \int_a^{\hat{t}} f(t)(p_1^f + tp_1^\ell - c_2(0,\hat{t}))dt \tag{84}$$

$$=F(\hat{t})c_2(0,\hat{t}) - F(a)c_2(0,a) \tag{85}$$

$$- (F(\hat{t}) - F(a))(p_1^f + \mu(a,t)p_1^\ell) \tag{86}$$

and for $a > \hat{t}$ the profit change is given by

$$F(\hat{t})(c_2(0,\hat{t}) - c_2(0,a)) \tag{87}$$

$$+ \int_{\hat{t}}^{a} f(t)(p_1^f + tp_1^\ell c_2(0,a))dt \tag{88}$$

$$=F(\hat{t})c_2(0,\hat{t}) - F(a)c_2(0,a) \tag{89}$$

$$- (F(\hat{t}) - F(a))(p_1^f + \mu(a,t)p_1^\ell). \tag{90}$$

Bounding all of these expressions above by zero yields the first half of the theorem.

Equivalently, we now consider how the profit changes when provider 1 obtains the low variance users (i.e. $[0,a]_{\to 1}$):

$$\int_0^a f(t)(p_1^f + tp_1^\ell - c_1(0,a))dt \tag{91}$$

$$- \int_{\hat{t}}^{t_{max}} f(t)(p_1^f + tp_1^\ell - c_1(\hat{t},t_{max}))dt \tag{92}$$

$$=F(a)(p_1^f + \mu(0,a)p_1^\ell - c_1(0,a)) \tag{93}$$

$$- (1-F(\hat{t}))(p_1^f + \mu(\hat{t},t_{max})p_1^\ell - c_1(\hat{t},t_{max})) \tag{94}$$

$$\leq 0 \tag{95}$$

and

$$\int_a^{t_{max}} f(t)(p_1^f + tp_1^\ell - c_2(a, t_{max}))dt \tag{96}$$

$$- \int_0^{\hat{t}} f(t)(p_1^f + tp_1^\ell - c_2(0, \hat{t}))dt \tag{97}$$

$$=(1 - F(a))(p_1^f + \mu(a, t_{max})p_1^\ell - c_2(a, t_{max})) \tag{98}$$

$$- F(\hat{t})(p_1^f + \mu(0, \hat{t})p_1^\ell - c_2(0, \hat{t})) \tag{99}$$

$$\leq 0 \tag{100}$$

If the providers are symmetric, both providers would get the same profit if they served the same segment of the market, and therefore the conditions $(79) - (81)$ and $(84) - (86)$ do not have to be checked. □

References

Baye and Kovenock, 2017. Michael R. Baye and Dan Kovenock. *Bertrand Competition*, pages 1–7. Palgrave Macmillan UK, London, 2017.

Blattberg and Wisniewski, 1989. Robert C Blattberg and Kenneth J Wisniewski. Price-induced patterns of competition. *Marketing Science*, 8(4):291–309, 1989.

Celebi and Fuller, 2012. Emre Celebi and J David Fuller. Time-of-use pricing in electricity markets under different market structures. *IEEE Transactions on Power Systems*, 27(3):1170–1181, 2012.

Cramton, 2017. Peter Cramton. Electricity market design. *Oxford Review of Economic Policy*, 33(4):589–612, 2017.

Dierks and Seuken, 2019. Ludwig Dierks and Sven Seuken. Cloud pricing: The spot market strikes back. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. ACM, 2019.

Dierks *et al.*, 2019. Ludwig Dierks, Ian A. Kash, and Sven Seuken. On the cluster admission problem for cloud computing. In *Proceedings of the 14th Workshop on the Economics of Networks, Systems and Computation*, June 2019.

Ericsson, 2017. Ericsson. Shifting mobile data plans, 2017.

Feng *et al.*, 2013. Yuan Feng, Baochun Li, and Bo Li. Price competition in an oligopoly market with multiple iaas cloud providers. *IEEE Transactions on Computers*, 63(1):59–73, 2013.

Gallego *et al.*, 2006. Guillermo Gallego, Woonghee Tim Huh, Wanmo Kang, and Robert Phillips. Price competition with the attraction demand model: Existence of unique equilibrium and its stability. *Manufacturing & Service Operations Management*, 8(4):359–375, 2006.

Joskow and Wolfram, 2012. Paul L Joskow and Catherine D Wolfram. Dynamic pricing of electricity. *American Economic Review*, 102(3):381–85, 2012.

López-Pérez *et al.*, 2009. David López-Pérez, Alvaro Valcarce, Guillaume De La Roche, and Jie Zhang. Ofdma femtocells: A roadmap on interference avoidance. *IEEE Communications Magazine*, 47(9):41–48, 2009.

Moorthy, 1984. K Sridhar Moorthy. Market segmentation, self-selection, and product line design. *Marketing Science*, 3(4):288–307, 1984.

Muratori and Rizzoni, 2015. Matteo Muratori and Giorgio Rizzoni. Residential demand response: Dynamic energy management and time-varying electricity pricing. *IEEE Transactions on Power systems*, 31(2):1108–1117, 2015.

Mussa and Rosen, 1978. Michael Mussa and Sherwin Rosen. Monopoly and product quality. *Journal of Economic Theory*, 18(2):301 – 317, 1978.

Rong *et al.*, 2018. Jiang Rong, Tao Qin, and Bo An. Dynamic pricing for reusable resources in competitive market with stochastic demand. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Sinha *et al.*, 2017. Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2017.

Truong-Huu and Tham, 2014. Tram Truong-Huu and Chen-Khong Tham. A novel model for competition and cooperation among cloud providers. *IEEE Transactions on Cloud Computing*, 2(3):251–265, 2014.

Urieli and Stone, 2016. Daniel Urieli and Peter Stone. Autonomous electricity trading using time-of-use tariffs in a competitive market. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Varian, 1989. Hal R Varian. Price discrimination. *Handbook of Industrial Organization*, 1:597–654, 1989.

5 Revenue Maximization for Consumer Software: Subscription or Perpetual License?

The content of this chapter has previously appeared in:

> Ludwig Dierks and Sven Seuken (2020) **Revenue Maximization for Consumer Software: Subscription or Perpetual License?**. Working Paper.

Revenue Maximization for Consumer Software: Subscription or Perpetual License?

Ludwig Dierks and Sven Seuken

Department of Informatics, University of Zurich
{dierks,seuken}@ifi.uzh.ch

Abstract. We study the revenue maximization problem of a publisher selling consumer software. We assume that the publisher sells either traditional perpetual licenses, subscription licenses, or both. For our analysis, we employ a game-theoretic model, which enables us to derive the users' equilibrium strategies and the publisher's optimal pricing strategy. Via extensive numerical evaluations, we then demonstrate the sizable impact different pricing strategies have on the publisher's revenue, and we provide comparative statics for the most important settings parameters. Although in practice, many publishers still only sell perpetual licenses, we find that offering a subscription license in addition to a perpetual license typically (but not always) leads to significantly higher revenue than only selling either type of license on its own.

Keywords: Revenue Management · Pricing · Consumer Software· Subscription · Product Differentiation

1 Introduction

Consumer software, particularly video games, is a multi-billion dollar industry [16, 10]. Originally sold on physical media like CDs or DVDs, the rise of fast network connections has allowed software markets to become increasingly digital, eschewing any physical medium. This has brought with it a proliferation of new business models such as microtransactions (i.e., the sale of many mini-upgrades for small amounts of money), lootboxes (i.e., randomized microtransactions [3]) or in-game advertisement [1].

In this paper, we analyze the revenue maximization problem of a software publisher who, while still focused on selling *licenses* for his product, is open to do this either in the form of perpetual or subscription licenses. Whereas a classic perpetual license, once bought, allows a user access to the product for as long as he desires (or, in some more recent cases, as long as the publisher supports it), a subscription license only allows access to the product for as long as the user pays a (typically monthly) recurring fee.[1] While in recent years, subscription licenses have become common for cloud-based Software-as-a-Service offerings (where their main selling point is access to cloud hardware), most products that do not come with significant cloud hardware are still only

[1] This is distinct from subscription services that give access to constantly changing bundles of products (e.g. Xbox Game Pass).

sold through perpetual licenses (though some publishers have recently experimented with subscription models [11, 12]). Since we are interested in the revenue effects of selling the product itself (as opposed to additional cloud-based features), we exclude any cloud-based synergy from our analysis.

For a publisher, offering a subscription model has a few obvious advantages compared to perpetual licenses: the barrier of entry gets reduced and natural product differentiation takes place between users depending on how long they are interested in the product. This potentially allows the publisher to obtain far higher revenue from some users than he could with perpetual licenses and may allow him to support the product for a longer period of time through a continuous revenue stream.

On the other hand, offering a subscription model also comes with certain disadvantages. Users may stop subscribing once a product's novelty fades, while some users that would use the product for a long time with low intensity may not be willing to pay a recurring price at all. In addition, if the option to alternatively buy a perpetual license is also offered, "market cannibalization" between both offerings may occur. And lastly, but importantly, while publishers traditionally sell upgrades to keep their product up to date or expand its features, with a subscription model it is typically assumed that users always obtain access to the most recent version (not including optional microtransactions).

In this paper, we take a game-theoretic approach towards analyzing the merit of offering subscription licenses instead of or in addition to perpetual licenses. Selling a product over some time horizon is fundamentally a question of revenue management [6, 4, 14] and it is important to take user behavior into account, as users for example may delay a purchase to wait for a reduction in prices. In contrast to classic revenue management problems, software as a purely digital good has neither a limited stock nor marginal costs. Instead, the quality of the product in the eyes of users continuously decays [7]. Furthermore, offering a subscription option and offering paid but optional upgrades both constitute forms of product differentiation [9, 8, 5], though again with very particular cost and utility structures that differ from the classic literature. In the past, the revenue effects of subscriptions have been studied for some other domains like ancillary services of a repeatedly sold core product (e.g., additional baggage for airline tickets) [17] or professional Software-as-a-Service offerings (where, importantly, subscriptions provide scalable hardware while buy options do not, and utilities take a very different form than for consumer software) [13]. Chawla et al. [2] effectively studied a kind of subscription service with a free trial period for software products, though they restricted themselves to a single user type and evaluated their mechanism compared to extracting all expected value from the user.

To properly analyze the problem and capture all its particularities, we introduce a tailor-made model that takes the form of a two-step game. In the first step, the publisher chooses his pricing strategy; in the second step, the users act inside of a discrete time sub-game where they arrive and dynamically obtain and lose demand for the product. We prove that there are only five distinct classes of user equilibrium strategies, which significantly aids in our analysis. Based on this, we derive the publisher's revenue as a function of his pricing strategy and show that only offering a subscription option can never be optimal. We show that depending on the setting, either only offering

perpetual licences or offering a subscription option in addition to perpetual licenses can be optimal, though for most software products, offering both options is likely to lead to the best possible revenue. Through comparative statics we further evaluate the influence different setting parameters have on the revenue of the various pricing strategies.

2 Model

We model the problem as a two stage game. First, the publisher commits to a pricing strategy. Then, over n_{max} timesteps users arrive to the system. Once a user has arrived to the system, he faces a game with multiple timesteps. We can model this as an infinite time horizon Markov Decision Process (MDP) where he can take actions and obtains rewards depending on the state he is in.

2.1 Publisher Model

The publisher wants to sell a digital product. After m timesteps, he offers an optional upgrade to the product.[2] The product has a base quality q_1, which the upgrade additively increases by q_2. While the base product is always available, the upgrade only becomes available from timestep m onwards. Because of the digital nature of the good, we assume that the publisher has infinite supply and no marginal costs. The publisher's strategy space consists of his choice of price vector $p = (p_1^{<m}, p_1^{\geq m}, p_2, p_S)$. Thus, he offers a menu of options to his users: (1) buy the base product for a one-time payment $p_1^n = p_1^{<m}$ (if bought in timestep $n < m$) or (2) for $p_1^n = p_1^{\geq m}$ (if bought in timestep $n \geq m$), (3) buy the upgrade for a one-time payment p_2 (only offered in timesteps $n \geq m$), or (4) subscribe for price p_S per time step. A product that is bought can be used forever, but the upgrade needs to be bought separately. On the other hand, a subscription gives immediate access to all available upgrades, but the user loses access to the product when his subscription lapses.

A publisher can choose to not offer a buy or subscribe option by setting the corresponding price to infinity. The publisher's utility is equal to his expected revenue per user.

2.2 User Model

Users are identified by their state and type. A user's state is given by a tuple $\sigma = (d, o)$. While users arrive interested in obtaining access to the product, after using the product for some time this interest may vanish. The demand $d \in \{0, 1\}$ denotes whether the user is still interested, i.e., whether he obtains any utility for having access to the product. The ownership vector $o \in \{0, 1\}^2$ denotes whether a user owns the base product, i.e., $o_1 = 1$ or the upgrade, i.e., $o_2 = 1$.

A user's type is a tuple $\tau = (n_a, \delta, \gamma, v)$. $n_a \in \{1, \ldots, n_{max}\}$ denotes the timestep the user arrives into the system, i.e., the earliest time he could buy the product and

[2] We limit ourselves in the analysis to exactly *one* upgrade after exactly m time steps to simplify the exposition. Turning m into a strategic variable, as well as extending our model to more than one upgrade or multiple price changes is straightforward.

is drawn from a distribution with pmf f_a. $\delta \in (0,1)$ denotes the user's long term engagement factor with the product and is drawn from a distribution with pmf f_δ. While any arriving user starts with demand $d = 1$ for the product, in any timestep in which he uses the product he has a probability of $1 - \delta$ to become uninterested and lose demand (setting $d = 0$). A user who has lost demand no longer obtains utility from having access to the product. The release of the upgrade has the complementary probability δ to rekindle a lapsed user's interest and set $d = 1$. $\gamma \in (0,1)$ denotes the quality decay factor of the user and is drawn from a distribution with pmf f_γ. While the product and the upgrade have qualities q_i in the timesteps in which they become available, the realized quality of the product for the users decreases every timestep as hype and novelty fade and it slowly becomes outdated. The *realized quality* of having access to o at time n is given by $q(o,\gamma,n) = \mathbb{1}_{[o_1=1]}(\gamma^n q_1 + \mathbb{1}_{[o_2=1]}\gamma^{n-m}q_2)$. Lastly, $v \in [0, v_{max}]$ denotes the value a user has for a product of quality 1, as long as he has demand. v is drawn from a distribution with pdf f_v.

A user's action space in any timestep consists of whether he subscribes $S \in \{0,1\}$ or buys the product or the upgrade $b \in \{0,1\}^2$. Subscription gives a user immediate access to anything currently released, i.e., $o_S^n = [1,0]$ if $n < m$ and $o_S^n = [1,1]$ otherwise, while buying b gives ownership of the bought product, i.e., changes the ownership vector from o to $o' = max(o,b)$.

The *normalized immediate reward* w_n of a user of type τ in timestep n and state (d,o) is given by $w_n(S,b,\tau,\sigma) = d\left((1-S)q(max(o,b),\gamma,n) + Sq(o_S^n,\gamma,n)\right)$, while his *immediate payment* is given by $\rho_n(S,b,\tau,\sigma,p) = p_S S + p_1^n b_1 + p_2 b_2$. His overall *immediate utility* in timestep n is therefore given by $u_n(S,b,\tau,\sigma,p) = vw_n(S,b,\tau,\sigma) - \rho_n(S,b,\tau,\sigma,p)$.

A strategy $\alpha(n,\sigma) : \mathbb{N} \times \{0,1\} \times \{0,1\}^2 \to \{0,1\} \times \{0,1\}^2$ maps timesteps and user states to actions. The *normalized expected reward* for playing strategy α is given by

$$w(\alpha,\tau) = \sum_{n=n_a}^{\infty} \sum_{\sigma'} P(\sigma_n = \sigma' | \alpha,\tau,\sigma) w_n(S,b,\tau,\sigma), \tag{1}$$

where $P(\sigma_n = \sigma' | \alpha,\tau,\sigma)$ denotes the probability of the user being in state σ' during timestep n given α,τ,σ. Similarly, the expected payment is given by

$$\rho(\alpha,\tau,p) = \sum_{n=n_a}^{\infty} \sum_{\sigma'} P(\sigma_n = \sigma' | \alpha,\tau,\sigma) \rho_n(S,b,\tau,\sigma,p). \tag{2}$$

A user's overall *expected utility* with strategy α is consequently given by $u(\alpha,\tau) = vw(\alpha,\tau) - \rho(\alpha,\tau,p)$.

3 User Equilibrium Strategies

Before we can analyze the publisher's revenue, we first need to determine how users would react to any given publisher strategy. Since the supply of software is unlimited and since we do not model any social effects, the utility of a given user is independent of the strategies of the other users. We can therefore find the equilibrium strategy of each user by solving his individual MDP in isolation.

For any given user type τ, we could directly do this through backward induction. But doing so on a user's full strategy space is computationally very costly, even for one user. As we later need to compute the optimal strategies for each user type to calculate the publisher's revenue, we now show that the optimal strategies for each user type can only come from a small set of possible strategies. Note that throughout this section, most expressions depend on the price p. For the sake of readability, we keep this dependency implicit and omit p wherever doing so does not cause confusion.

A first important observation for identifying potentially optimal user strategies is that conceptually, the MDP for each user type can be seen to consist of two distinct parts: anything that happens before the upgrade is released in timestep m and anything that happens afterwards. For notational clarity and ease of exposition, we split user strategies in this manner, i.e., setting $\alpha = (\alpha_1, \alpha_2)$, where α_1 denotes a user's strategy before m and α_2 denotes his strategy beginning from timestep m.

Before timestep m, a user's actions are restricted to buying the base product, doing nothing or subscribing. Note that subscribing in any timestep yields the same immediate reward as owning the base product. Given this, we easily obtain the following result that shows that there is only one potentially optimal strategy that involves buying the product and one potentially optimal strategy that involves subscribing (though possibly for zero timesteps). Overloading notation, we denote these by $\alpha_1 = b$ and $\alpha_1 = s$, respectively.

Lemma 1. *1. For a user of type $\tau = (n_a, \delta, \gamma, v)$ that buys the base product before timestep m, the optimal strategy $\alpha_1 = b$ has him buy in the timestep n_a he arrives and not subscribe in any timestep $n < m$.*

2. For a user of type $\tau = (n_a, \delta, \gamma, v)$ that does not buy the base product before timestep m and plays some α_2 from timestep m onwards, there exists $n_1^{\alpha_2,\tau}$ such that the optimal strategy $\alpha_1 = s$ has him subscribe in any timestep $n < n_1^{\alpha_2,\tau}$ where he has demand and no other timesteps. It holds that $n_1^{\alpha_2,\tau}$ is the smallest $n \geq 0$ for which it holds

$$v < \begin{cases} \frac{p_S - (1-\delta)^2 \rho((\alpha_s,\alpha_2),\tau',p)}{q([1,0],\gamma,n) - (1-\delta)^2 w((\alpha_s,\alpha_2),\tau')} & \text{if } n < m \\ \infty & \text{if } n = m \end{cases} \tag{3}$$

where $\tau' = (m, \delta, \gamma, v)$ (i.e., $\rho((\alpha_s, \alpha_2), \tau', p)$ and $w((\alpha_s, \alpha_2), \tau')$ are the reward and payment the user would obtain if he arrived in timestep m).

Proof. The first statement follows directly by noting that when buying in a later timestep before m, a user's additional realized value compared to subscribing or doing nothing decreased, but his payment does not. He therefore buys in the first possible timestep and afterwards can not obtain any additional reward through subscribing.

For the second statement, note that a subscribing user obtains immediate utility $vq([1,0], \gamma, n) - p_S$ in any subscribed timestep, which decreases in n. Thus, if he does not subscribe in any timestep $n_1^{\alpha_2,\tau}$, then he will also not subscribe in any later timestep before m. Additionally, while subscribed, he has probability $(1 - \delta)$ to lose demand. If he loses demand before timestep m, then he still has probability δ to regain demand in timestep m. This means that subscribing in timestep $n < m$ decreases the expected utility he obtains after timestep m by a factor of δ^2. As he does not buy before m, his

utility from timestep m onward, if $d = 1$, is the same as if he had arrived in timestep m, i.e., as if his type was $\tau' = (m, \delta, \gamma, v)$. The overall change in expected utility for subscribing in the largest timestep n he subscribes is thus given by

$$\begin{aligned} & vq([1,0],\gamma,n) - p_s - v(1-\delta)^2 w((\alpha_s,\alpha_2),\tau') \\ & + (1-\delta)^2 \rho((\alpha_s,\alpha_2),\tau',p) \end{aligned} \tag{4}$$

and $n_1^{\alpha_2,\tau}$ is the first timestep for which this change would be negative.

Given Lemma 1 and a strategy α_2 to play from timestep m onwards, there are only two potentially optimal strategies before timestep m: buy in the timestep a user arrives, denoted by $\alpha_1 = b$, or subscribe until timestep $n_1^{\alpha_2,\tau}$, denoted by $\alpha_1 = s$. To determine which of these two strategies is optimal, we must next take the user's strategy after the upgrade release into account.

After the upgrade releases, the user's space of potentially optimal strategies grows slightly. In addition to buying the upgrade, and if not yet owned, the base product ($\alpha_2 = b$) or not buying anything ($\alpha_2 = s$), a user might also decide to only buy the base product, but not the upgrade ($\alpha_2 = b_b$). This can for example happen if the base product is heavily discounted after timestep m, but the price of the upgrade is set very high. For such a user it might be optimal to first subscribe for a few timesteps, before buying the base product.

Lemma 2. *1. For a user of type τ who buys the upgrade (and, if not yet owned, the base product), the optimal strategy $\alpha_2 = b$ is to buy as soon as possible (i.e., in timestep $max(n_a, m)$) and to not subscribe in any timestep $n \geq m$.*

2. For a user of type τ with ownership vector o who does not buy anything after timestep m, for the optimal strategy $\alpha_2 = s$ there exists a timestep $n_2^{o,\tau} \geq m$ such that he subscribes in any timestep n with $m \leq n < n_2^{o,\tau}$ where he has demand $d = 1$ and subscribes in no timestep $n \geq n_2^{o,\tau}$. It holds that $n_2^{o,\tau}$ is the smallest $n \geq 0$ for which

$$v < \frac{p_S}{q([1,1]-o,\gamma,n)}. \tag{5}$$

3. For a user of type τ that only buys the base product after timestep m (and never buys the upgrade), for the optimal strategy $\alpha_2 = b_b$ there exists a timestep n_3^τ such that he subscribes in and only in any timestep n with $m \leq n < n_3^\tau$ where he has demand and buys in timestep n_3^τ (if he still has demand). It holds that n_3^τ is the smallest $n \geq 0$ for which

$$v < \frac{p_S - (1-\delta)p_1^{\geq m}}{q([0,1],\gamma,n_3^\tau)}. \tag{6}$$

Proof. 1. and 2. follow analogously to Lemma 1. To see 3., note that a user who does not buy the upgrade, but does buy the base product after timestep m, optimally does so in the first timestep where he does not subscribe. Not doing so only decreases his reward but not his payment. When subscribing in timestep n, such a user obtains an additional value of $q([0,1],\gamma,n)v$ over just owning the base product and makes a

payment of p_S. He has a probability of $(1-\delta)$ to lose demand before the next timestep, in which case he does not buy at all, saving (in expectation) $(1-\delta)p_1^{\geq m}$. Thus, a user's utility for subscribing in the highest timestep in which he subscribes is given by $q([0,1],\gamma,n)v - p_S + (1-\delta)p_1^{\geq m}$. As this utility decreases in n, the user either does not subscribe at all (i.e. buys in timestep m or when he arrives) or there exists a smallest timestep $n_3^{\gamma,v}$ for which his added utility for subscribing becomes negative and in which he buys. Note that when buying in timestep $n_3^{\gamma,v}$ is not rational, then playing $alpha_2 = b_b$ is dominated by $\alpha_2 = s$.

Given Lemma 2, there are only three potentially optimal strategies beginning in timestep m: (1) buy the upgrade (and, if not owned yet, the base product) once the upgrade releases in m (or once the user arrives if $n_a > m$), (2) not buy anything and subscribe until timestep $n_2^{o,\tau}$, or (3) subscribe before timestep n_3^{τ}, then buy the base product in time step n_3^{τ}. We denote these by $\alpha_2 = b$, $\alpha_2 = s$ and $\alpha_2 = b_b$.

Taken together, Lemmas 1 and 2 describe all potentially optimal strategies for any user.

Proposition 1. *It maximizes the expected utility of a user of type τ to play strategy*

$$\begin{aligned}\alpha^*_{\tau,p} =& argmax_{\alpha_1\in\{b,s\},\alpha_2\in\{b,s,b_b\}} vw((\alpha_1,\alpha_2),\tau) \\ & - \rho((\alpha_1,\alpha_2),\tau,p)\end{aligned} \tag{7}$$

Proof. Follows from combining Lemmas 1 and 2.

4 Publisher Revenue

Given the results for the optimal user strategies from Section 3, we can now give a relatively simple expression for the publisher's revenue.

Proposition 2. *Given strategy $p = (p_1^{<m}, p_1^{\geq m}, p_2, p_S)$, the publisher's expected revenue per user is given by*

$$\begin{aligned}\pi(p) =& \sum_{n_a} f_a(n_a) \sum_{\delta} f_\delta(\delta) \sum_{\gamma} f_\gamma(\gamma) \\ & \int_v \rho\left(\alpha^*_{(n_a,\delta,\gamma,v),p}, (n_a,\delta,\gamma,v), p\right) f_v(v) dv\end{aligned} \tag{8}$$

where $\alpha^*_{(n_a,\delta,\gamma,v),p}$ is given by Proposition 1

Proof. Follows directly by taking the optimal user strategies $\alpha^*_{(n_a,\delta,\gamma,v),p}$ for each type τ as given by Proposition 1 and taking the expectation over all types.

4.1 Optimality Results

We now state a few general theoretical insights. First, only offering a buy option is optimal in at least some settings, i.e., also offering a subscription option can not always be used to increase revenue. Secondly, *only* offering a subscription option can never be as good as offering both options to users.

Theorem 1. *There exist settings where setting $p_S = \infty$, i.e., only offering a perpetual licenses, maximizes $\pi(p)$. There exists no setting where setting $p_1^{<m} = p_1^{\geq m} = p_2 = \infty$, i.e., only offering subscription licenses, maximizes $\pi(p)$.*

Proof. To see that only offering a buy option can be optimal, consider a setting without a later upgrade (i.e., $m = 1$) and only a single user type τ. Since the expected reward for owning is higher than the expected reward for subscribing up to any finite timestep, users are willing to pay more for perpetual licenses than for subscribing. Since there is only one user type and by Lemma 2 buying users buy in the timestep they arrive, adding an additional subscription option can not extract additional revenue.

To see that only offering a subscription option can never be optimal, assume some p with $p_S < \infty$ and $p_1^{<m} = p_1^{\geq m} = p_2 = \infty$ is optimal. Given p, let τ_{max} be the set consisting of the user types that make the highest expected payment after timestep m. Denote this payment by $\rho_{max}^{>m}$. Since the reward for buying in timestep m is always strictly higher than the reward for subscribing from m to any finite timestep, users of type τ_{max} would be willing to pay $\rho_{max}^{>m} + \epsilon$ for owning the product from timestep m onwards. Setting $p_1^{\geq m} = \rho_{max}^{>m} + \epsilon$, $p_2 = 0$ for $\epsilon > 0$ small enough therefore leads to users in a neighborhood around τ_{max} buying in timestep m and paying strictly more than $\rho_{max}^{>m}$, while no user pays less. Consequently, $p = (\rho_{max}^{>m} + \epsilon, \infty, 0, p_S)$ yields higher revenue for the publisher and only offering subscriptions cannot be revenue optimal.

Importantly, this does not make any statement about whether only offering an (optimal) buy or subscription option would yield a better revenue and we can find settings where either yields a higher revenue. Unfortunately, there does not seem to be a simple condition for deciding which option is optimal or by how much revenue could increase for offering both options simultaneously without solving for the optimal pricing strategy p.

4.2 Calculating Optimal Prices and Revenue

In this section we discuss how to effectively derive optimal pricing strategies and revenue. Unfortunately, the publisher's revenue is highly non-convex in p and has many local maxima: Changing prices affects different users differently and usually increase the payments of some users but reduces the payment of other users. Consequently, first-order derivative tests are very bad indicators for whether a given price vector p is close to optimal. To still obtain (approximately) optimal strategies, we therefore have to employ global non-linear solvers. Fortunately, the search space is bounded: it is not reasonable to allow publishers to set negative prices and any price above the highest utility any user could obtain is effectively infinite.

While the expression for the publisher's revenue given by Proposition 2 can already be numerically evaluated, doing so is quite costly, making a search for the optimal strategy practically infeasible. We take two steps to rectify this. First, we derive simple closed form expressions for the normalized expected reward and the expected payment for all potentially optimal strategies a given user could play. Due to space constraints, these expression are found in Technical Appendix A.

Second, we get around having to do a numerical integration by noting that the payments are piecewise constant in the value v.

Proposition 3. *There exist (unique)* $v_1, \ldots, v_k$ *with* $v_1 = 0$, $v_i < v_{i+1}$ *and* $v_k = v_{max}$ *such that*

$$\pi(p) = \sum_{n_a} f_a(n_a) \sum_{\delta} f_\delta(\delta) \sum_{\gamma} f_\gamma(\gamma) \sum_{i=1}^{k-1} \rho\left(\alpha^*_{(n_a,\delta,\gamma,v^i),p}, (n_a, \delta, \gamma, v^i), p\right) \left(F_v(v^{i+1}) - F_v(v^i)\right) \tag{9}$$

A formal proof, which also illustrate how to derive the v_i, is given in Technical Appendix B. Broadly, this follows from the fact that, for fixed δ, γ and n_a and $\alpha \in \{b, s\} \times \{b, s, b_b\}$, the normalized expected reward and payment are piecewise constant in v.

As the expected revenue can now be evaluated relatively cheaply, an (approximately) optimal solutions can easily be found by for example employing differential evolution search [15], a stochastic method where a population of random candidate solutions is stochastically mutated until convergence is archived. As a stochastic search, true global optimality of the found solution cannot be guaranteed. While even single runs with relatively small populations usually yield good results, there is a small probability to end up in a local maximum. It therefore is sensible to repeat the search a few times to minimize the optimality gap.

5 Numerical Evaluation

To better understand when offering subscriptions can increase a publisher's revenue and by how much it typically does so, we now present a number of numerical examples and comparative statics. For each example, we give the optimal revenue for the *optimal* prices for a publisher who either (1) only offers a buy option (Opt(Buy)), (2) only offers a subscription option (Opt(Sub)), (3) offers both options (Opt(Both)), or (4) offers both options, but restricts the buy prices of perpetual licenses to those that would be optimal without subscription option (Opt(Both | Opt(Buy))).

5.1 Example Domain: Video Games

For our numerical analysis to have merit, we need to choose realistic settings parameters. We chose the domain of video games for our numerics, because some user data as well as pricing data is available for this domain, which we can use to inform our choice of parameters and distributions.[3] We base our distributions on a dataset obtained from the website *Steam Spy* [4], which collects publicly available data from the large video game storefront *Steam*[5] to statistically estimate the number of owners of video games over time and what percentage of them actively used the game recently. In the following, we describe the distributions and parametrizations we choose based on insights from

[3] To the best of our knowledge, no comprehensive study of how users in this domain are typically distributed is available, which is why we have to base our parameter choices on a rough analysis of some limited datasets that we have access to.

[4] https://steamspy.com/about

[5] https://store.steampowered.com/about/

	$p_1^{<m}$	$p_1^{\geq m}$	p_2	p_S	Revenue	User Welfare	Overall Welfare
Opt(Buy)	45.82	21.8	18.06	∞	31.42	33.23	64.65
Opt(Sub)	∞	∞	∞	14.66	33.54	24.07	57.61
Opt(Both)	96.98	35.19	47.96	17.71	37.88	24.39	62.27
Opt(Both | Opt(Buy))	45.82	21.8	18.06	18.4	31.82	33.89	65.71

Table 1. Results for four different publisher strategies for the base case

this dataset. A discussion of the data and how we determined our parameters can be found in Technical Appendix C, while a snapshot of some representative games can be found in Technical Appendix D.

We set the product's base quality to $q_1 = 1$ and the quality of the upgrade to $q_2 = 0.5$. We assume that the upgrade releases in timestep $m = 6$ and that the publisher stops to actively sell the product in timestep $n_{max} = 12$ (after which only user's that have bought it can continue to use it).

Denoting the overall arrivals during the first timestep by x_a, we set the arrival distribution f_a to

$$f_a(n_a) = \begin{cases} \frac{x_a}{x_a + n_{max} - 1} & \text{if } n_a = 1 \\ \frac{1}{x_a + n_{max} - 1} & \text{if } 1 < n_a < n_{max} \end{cases} \tag{10}$$

While we vary x_a for our comparative statics, we choose $x_a = 5$ as the standard value for most of the section.

We assume that $\gamma \in \{0.85, 0.9, 0.95\}$. We let x_γ denote the probability that γ is 0.9 and set the γ distribution f_γ to be $f_\gamma(0.9) = x_\gamma$ and $f_\gamma(0.85) = f_\gamma(0.95) = \frac{1}{2}(1 - x_\gamma)$. While we vary x_γ for our comparative statics, we choose $x_\gamma = 0.8$ as the standard value for most of the section.

We assume that 20% of users have a 90% probability to not lose demand in each timestep, i.e., $\delta = 0.9$. We therefore settle on a simple two-type distribution of long-term and short-term users. Denoting the probability that a short-term user does not lose demand after one timestep by x_δ, we obtain $f_\delta(x_\delta) = 0.8$ and $f_\delta(0.9) = 0.2$. While we vary x_δ for our comparative statics, we choose $x_\delta = 0.5$ as the standard value used for most of the section.

We assume that the user values are distributed according to a normal distribution with mean $\mu = 25$ truncated to $[0, 50]$. While we vary the standard deviation σ of f_v for our comparative statics, we choose $\sigma = 10$ as the standard value for most of the section.

All optimal publisher strategies are calculated using a best-of-15 differential evolution search. While full optimality for all data points cannot be guaranteed, the magnitude of any remaining approximation error is insignificant for any general insights.

5.2 Base Case

In this section, we discuss the numerical results for the base case with $x_a = 5, x_\gamma = 0.8, x_\delta = 0.5$ and $\sigma = 10$. This parametrization roughly corresponds to a typical game

based on our analysis of the Steam Spy data. The results for each type of publisher strategy are summarized in Table 1.

The best attainable revenue for a publisher who only wants to sell perpetual licenses without offering a subscription option (i.e., Opt(Buy)) is $\pi(p) = 31.42$. As this revenue is attained with a relatively low price (that is further discounted roughly 50% once the upgrade releases), the publisher ensures that most users buy his product. This leaves the users with relatively high utility for owning the game and thus the users' social welfare (i.e., the expected utility of a randomly drawn user) is relatively high at 33.23.

If the publisher instead only offers a subscription option (i.e., Opt(Sub)), his expected revenue per user increases to $\pi = 33.54$, a substantial 6.7% increase over only offering the buy option. This is possible because, when only offering a buy option the publisher had to set a relatively low price to attract users that are only interested in playing the game for a short time as well as users that want to play it for a long time. The subscription option on the other hand automatically price discriminates between those user types and extracts more revenue from long-term users. Consequently, this revenue increase comes at the cost of the user welfare, which decreases substantially. Unfortunately, this is not simply a transfer of utility from the users to the publisher, as the users welfare decreases more than the publisher's revenue increases. This loss is caused because users whose perceived quality of the game decayed too much stop subscribing, even though they would like to continue playing. They are simply not willing to pay the subscription price anymore. This decreases the system's overall welfare (i.e., the sum of revenue and user welfare) by 11%.

Offering both perpetual and subscription licenses (i.e., Opt(Both)), the publisher he can increase his revenue to $\pi = 37.87$, an additional 12.9% increase over only offering a subscription option and a staggering 20.5% increase over only selling perpetual licenses. This revenue increase requires that all prices rise compared to when only one of the two license types are offered. While the subscription price only moderately increases to $p_S = 17.71$, the buy prices roughly double to $p_1^{<m} = 98.98, p_1^{\geq m} = 35.19, p_2 = 47.96$. Interestingly, with these prices, any user that subscribes for 3 or less timesteps before the price change and then buys the discounted base product pays less overall than if he would have bought the base product directly. Consequently, we 52.9% of the users arriving in timestep 1 subscribe at first, but ultimately buy a perpetual licenses once it is discounted (as long as they still have demand). Only 15.3% of users that arrive in timestep 1 directly buy the game at its high starting price. This effectively splits the user base in two. First there are casual users that subscribe for a few timesteps and often pay less than they would when only perpetual licenses are offered. Then there are power users that plan to use the game for a long time, often longer than they would be willing to subscribe, that pay the increased buy price. Consequently, despite the notable revenue increase, user welfare does not decrease further and, compared to only offering a subscription option, even slightly increases to 24.4. While this is still notably lower than the user welfare when only offering perpetual licenses, the system's overall welfare (i.e., the sum of revenue and user welfare) now is only about 3.7% lower, showing that most of the user welfare gets transferred to the publisher instead of being lost.

Lastly, we analyze whether the publisher can increase his revenue by offering a subscription option without changing the prices of perpetual licenses (i.e., Opt(Both|

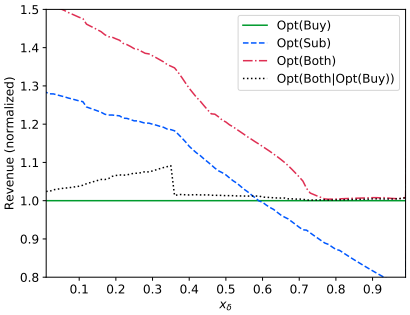

Fig. 1. Revenue for different long-term engagement distributions (i.e., varying x_δ)

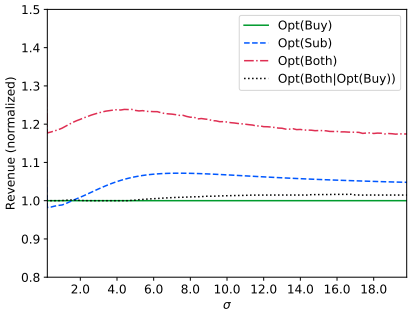

Fig. 2. Revenue for different value distributions (i.e., varying σ)

Opt(Buy))). This guarantees that no user is worse off than when only perpetual licenses are offered, an important consideration when there are competing products. Any revenue increase under such a pricing model has to come from additional users that do not buy perpetual licenses even when no subscription option is offered. Fixing the buy prices at $p_1^{<m} = 45.82, p_1^{\geq m} = 21.8, p_2 = 18.06$, the optimal subscription price is $p_S = 18.4$. Considering that the publisher wants to attract additional users, it might seem counterintuitive that p_S here is even higher than what was optimal combined with the far higher optimal buy price. This effect occurs because the lower the buy price, the more users with low long-term engagement buy when no subscription option is offered. But those same users readily switch to subscribing and pay even less when p_S is low, increasing market cannibalization. Consequently, the attainable revenue increase is comparatively modest, with the publisher obtaining at most $\pi = 31.82$, an increase of 1.3%. While this pales compared to the potential revenue increase with fully optimized prices, it can still constitute an additional revenue of hundredth of thousands or even millions of dollars for large releases. Importantly, since this pricing strategy, by construction, leads to a Pareto improvement for the users, user welfare also slightly increases.

5.3 Comparative Statics

We now study how varying the setting parameters x_δ and σ affects the publisher's revenue under his four different strategies. In Figures 1 and 2, we present comparative statics for how the optimal revenue of each type of pricing strategy changes in relation to the revenue of only offering a buy option (which we normalize to 1). Due to space constraints, similar comparative statics for x_γ and x_a can be found in Technical Appendix E. Both of them show a comparatively minor, but notable effect on the relative revenue potential.

In Figure 1, we see that the subscription option is best when the spread between short-term and long-term users is largest (i.e., x_δ is small), as subscription inherently differentiates users on how long they are interested in the product. For very high x_δ,

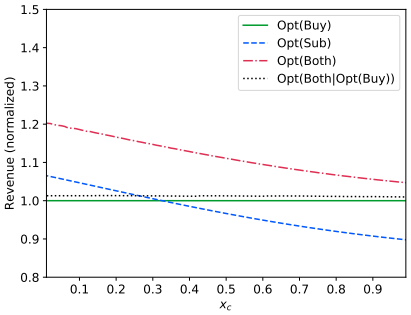

Fig. 3. Revenue for different levels of correlation (i.e., varying x_c)

i.e., when every user in expectation has demand for many timesteps, the potential revenue gain of offering a subscription alongside a buy option becomes very small (on the magnitude of 0.5%). At that point, too many users stop subscribing despite still having demand because their perceived value decayed below the subscription price. This makes subscription options inherently unattractive and any subscription price that could attract a large amount of additional users would cause too much market cannibalization. Interestingly, for low x_δ (< 0.36 when only offering a buy option, < 0.46 when offering both options), it is optimal to make the upgrade free (but increase the base product's price). This causes the sudden lage drop in attainable revenue increase for adding a subscription option without changing the buy price, as users with relatively high quality decay that were priced out of buying before that point (and thus got a relatively expensive subscription for 1 or 2 time steps) afterwards switch to only buying the base game.

In Figure 2, we see that the variance of the user values for the most part has a relatively low impact on the relative revenue potential of the different strategies. But since the optimal price when only offering a buy option for with the given parameters is low enough that users with average valuation buy even if the have a low long-term engagement factor δ and decay factor γ, the potential revenue advantage of offering a subscription option without changing buy prices goes to zero with low σ.

5.4 Correlating Value and Long Term Demand

So far all type variables were assumed to be independent from each other. In practice it is likely that some (but not all) users with high long term engagement factor δ have lower values v, for example because they do not have much leisure time to use the game in each timestep and therefore need to own it for longer to spend the same time using it. In this example we want to study how introducing some correlation between δ and v changes the publisher's revenue. To that end, we let the value distribution depend on δ. For a dependence factor of x_c set the value distribution for a users with long-term engagement factor δ to a normal distribution with mean $\mu = 25((1 - x_c) + x_c(1 - \delta))$ and standard deviation $\sigma = 10$, again truncated to $[0, 50]$. As we can see in Figure 3,

while increasing the dependence between value and long-term engagement decreases the revenue potential of a subscription option, offering both options is still markedly better even when the mean of the normal distribution underlying a users value is fully dependent on the user's long-term engagement factor (i.e., $x_c = 1$). Interestingly, this dependence does not seem to have much, if any, effect on the revenue potential of introducing a subscription option without changing the buy prices.

6 Conclusion

We have analyzed the revenue maximization problem of a publisher wanting to either sell perpetual or subscription licenses for consumer software. In conclusion, combining subscription and perpetual license is typically revenue optimal when selling consumer software, realistically increasing revenue by $10\% - 20\%$ over only offering perpetual license. Offering both types of licenses, it is often further possible to combine a revenue increase compared to only offering perpetual licenses with a Pareto improvement for the users, though the resulting revenue increase is then only on the magnitude of $1\% - 2\%$.

References

1. Burns, Z., Roseboom, I., Ross, N.: The sensitivity of retention to in-game advertisements: An exploratory analysis. In: Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference (2016)
2. Chawla, S., Devanur, N.R., Karlin, A.R., Sivan, B.: Simple pricing schemes for consumers with evolving values. In: Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms. p. 1476–1490. SODA '16, Society for Industrial and Applied Mathematics, USA (2016)
3. Chen, N., Elmachtoub, A.N., Hamilton, M., Lei, X.: Loot box pricing and design. In: Proceedings of the 21st ACM Conference on Economics and Computation. p. 291–292. EC '20, Association for Computing Machinery, New York, NY, USA (2020)
4. Chen, Y., Farias, V.F., Trichakis, N.: On the efficacy of static prices for revenue management in the face of strategic customers. Management Science **65**(12), 5535–5555 (2019)
5. Desai, P.S.: Quality segmentation in spatial markets: When does cannibalization affect product line design? Marketing Science **20**(3), 265–283 (2001)
6. Gallego, G., Van Ryzin, G.: Optimal dynamic pricing of inventories with stochastic demand over finite horizons. Management science **40**(8), 999–1020 (1994)
7. Mao, W., Zheng, Z., Wu, F., Chen, G.: Online pricing for revenue maximization with unknown time discounting valuations. In: IJCAI. pp. 440–446 (2018)
8. Moorthy, K.S.: Market segmentation, self-selection, and product line design. Marketing Science **3**(4), 288–307 (1984)
9. Mussa, M., Rosen, S.: Monopoly and product quality. Journal of Economic Theory **18**(2), 301 – 317 (1978)
10. newzoo: Global games market report (2019), https://newzoo.com/insights/trend-reports/newzoo-global-games-market-report-2019-light-version/
11. PC Gamer: Paradox is testing a subscription service for europa universalis 4 (2020), https://www.pcgamer.com/paradox-is-testing-a-subscription-service-for-europa-universalis-4/

12. PC Gamer: Ubisoft clarifies that trackmania is subscription-based (2020), https://www.pcgamer.com/ubisoft-says-trackmania-is-not-subscription-based-you-just-pay-for-it-multiple-times/
13. Rohitratana, J., Altmann, J.: Impact of pricing schemes on a market for software-as-a-service and perpetual software. Future Generation Computer Systems **28**(8), 1328–1339 (2012)
14. Schlosser, R., Walther, C., Boissier, M., Uflacker, M.: Data-driven inventory management and dynamic pricing competition on online marketplaces. In: IJCAI. pp. 5856–5858 (2018)
15. Storn, R., Price, K.: Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization **11**(4), 341–359 (1997)
16. SuperData: 2019 year in review digital games and interactive media (2020), https://www.superdataresearch.com/2019-year-in-review
17. Wang, R., Dada, M., Sahin, O.: Pricing ancillary service subscriptions. Management Science **65**(10), 4712–4732 (2019)

A Reward and payment with the potentially optimal strategies

In this section we derive expressions for the expected reward and payment for any of the potentially optimal strategies that can be evaluated relatively cheaply. Towards this, we first derive the following technical lemma.

Lemma 3. *Assume a user of type τ with arrival time $n_a < m$ that either, beginning in timestep $n' = n_a$, subscribes in any timestep before $n'' \leq m$ and takes no other action or buys the product in timestep $n' = n_a$ (in which case $n'' = m$). The expected normalized reward such a user obtains* before timestep *m is given by*

$$w^{<m}(n', n'', o_1) = q_1(\gamma^{n'} \frac{1 - (lt\gamma)^{max(0,n''-n')}}{1 - lt\gamma} + \gamma^{n''} o_1 \frac{1 - (lt\gamma)^{m-n''}}{1 - lt\gamma}) \tag{11}$$

The probability that such a user has demand $d = 1$ in timestep m is given by

$$\kappa(n', n'', o) = \delta^{n''-n'} + \delta(1 - \delta^{n''-n'}) \tag{12}$$

Similarly, assume a user of type τ with value $v = 1$ and ownership vector o that has demand in timestep $n' = max(n_a, m)$ and subscribes from n' in any timestep before $n'' \geq m$ and takes no other action. The expected value such a user obtains after timestep m is given by

$$w^{\geq m}(n', n'', o) = (\gamma^{n'} q_1 + \gamma^{n'-m} q_2) \frac{1 - (lt\gamma)^{max(0,n''-n')}}{1 - lt\gamma} \tag{13}$$

$$+ (\gamma^{n''} q_1 o_1 + \gamma^{n''-m} q_2 o_2) \frac{1}{1 - lt\gamma} \tag{14}$$

A user of type τ that has demand $d = 1$ in timestep n' and subscribes from n' in any timestep before $n'' \geq m$ where he still has demand and takes no other action makes an expected payment of

$$\rho_S(n', n'') = p_S \frac{1 - (lt)^{n''-n'}}{1 - lt} \tag{15}$$

Proof. Recall that the normalized reward of a user of type τ that follows some strategy α is given by

$$w(\alpha, \tau) = \sum_{n=n_a}^{\infty} \sum_{\sigma'} P(\sigma_n = \sigma' | \alpha, \tau) w_n(S, b, \tau, \sigma'). \tag{16}$$

For a user with ownership vector o that subscribes from $n' = n_a$ to $n'' - 1$ and takes no other action (i.e., o never changes), the expected normalized reward before timestep m is consequently given by

$$w^{<m}(n', n'', o_1) = \sum_{n=n_a}^{m-1} \sum_{\sigma'} P(\sigma_n = \sigma' | (n', n''), \tau) w_n(S, b, \tau, \sigma') \tag{17}$$

$$= \sum_{n=n_a}^{m-1} P(d_n = 1 | \alpha, \tau) w_n(S, b, \tau, (1, o)). \tag{18}$$

For any timestep n in which the user subscribes, the probability to loose demand is $1-\delta$ and it follows

$$\sum_{n=n_a}^{n''-1} P(d_n = 1|\alpha,\tau) w_n(S,b,\tau,(1,o)) = \sum_{n=n_a}^{n''-1} \delta^{n-n_a} w_n(S,b,\tau,(1,o)) \tag{19}$$

$$= \sum_{n=n_a}^{n''-1} \delta^{n-n_a}\gamma^n q_1 \tag{20}$$

$$= \sum_{n=n_a}^{n''} (\gamma\delta)^{n-n_a}\gamma^{n_a} q_1 \tag{21}$$

$$= \gamma^{n_a} q_1 \frac{1-(\gamma\delta)^{n''-n_a}}{1-\gamma\delta}. \tag{22}$$

Here, the last equality follows as a partial geometric series. Analogously, taking into account that for $o_1 = 0$ no more value is obtained, for the remaining timesteps it holds

$$\sum_{n=n''}^{m-1} P(d_n = 1|\alpha,\tau) w_n(S,b,\tau,(1,o)) = \gamma^{n''} o_1 q_1 \frac{1-(\gamma\delta)^{m-n''}}{1-\gamma\delta}. \tag{23}$$

and the statement for $w^{<m}(n',n'',o_1)$ follows.

The probability that such a user still has demand in timestep n'' is given by $\delta^{n''-n'}$. If he does, then he also still has demand in timestep m. If he on the other hand does not have demand anymore in timestep n'', then he still has a probability of δ to regain demand with the upgrade release and price change in timestep m. The statements for $w^{\geq m}(n',n'',o)$ and $\rho_S(n',n'')$ follow by analogous arguments.

For all potentially optimal α, we can now easily derive a user's reward and payment.

Proposition 4. *For strategy $\alpha = (\alpha_1,\alpha_2)$ with $\alpha_1 \in \{b,s\}$, $\alpha_2 \in \{b,s,b_b\}$, the normalized expected reward for playing α is*

$$w((b,b),\tau) = \begin{cases} w^{<m}(n_a,n_a,1) + \kappa(n_a,n_a,1) w^{\geq m}(m,m,[1,1]) & \text{if } n_a < m \\ w^{\geq m}(n_a,n_a,[1,1]) & \text{if } n_a \geq m \end{cases} \tag{24}$$

$$w((b,s),\tau) = \begin{cases} w^{<m}(n_a,n_a,1) + \kappa(n_a,n_a,1) w^{\geq m}(m,n_2^{1,\tau},[1,0]) & \text{if } n_a < m \\ w^{\geq m}(n_a,n_2^{1,\tau},[1,0]) & \text{if } n_a \geq m \end{cases} \tag{25}$$

$$w((s,s),\tau) = \begin{cases} w^{<m}(n_a,n_1,0) + \kappa(n_a,n_1^{s,\tau},0) w^{\geq m}(m,n_2^{0,\tau},[0,0]) & \text{if } n_a < m \\ w^{\geq m}(n_a,n_2^{0,\tau},[0,0]) & \text{if } n_a \geq m \end{cases} \tag{26}$$

$$w((s,b),\tau) = \begin{cases} w^{<m}(n_a,n_1^{b,\tau},0) + \kappa(n_a,n_1^{b,\tau},0) w^{\geq m}(m,m,[1,1]) & \text{if } n_a < m \\ w^{\geq m}(n_a,n_a,[1,1]) & \text{if } n_a \geq m \end{cases} \tag{27}$$

$$w((s,b_b)) = \begin{cases} w^{<m}(n_a,n_1^{b_b,\tau},0) + \kappa(n_a,n_1^{b_b,\tau},0) w^{\geq m}(m,n_3^{\tau},[1,0]) & \text{if } n_a < m \\ w^{\geq m}(m,n_3^{\tau},[1,0]) & \text{if } n_a \geq m \end{cases} \tag{28}$$

The expected payments are

$$\rho((b,b),\tau) = \begin{cases} p_1^{<m} + \kappa(n_a, n_a, 1)p_2 & \text{if } n_a < m \\ p_1^{\geq m} + p_2 & \text{if } n_a \geq m \end{cases} \tag{29}$$

$$\rho((b,s),\tau) = \begin{cases} p_1^{<m} + \kappa(n_a, n_a, 1)\rho_S(n_m, n_2^{1,\tau}) & \text{if } n_a < m \\ p_1^{\geq m} + \rho_S(n_a, n_2^{1,\tau}) & \text{if } n_a \geq m \end{cases} \tag{30}$$

$$\rho((s,s),\tau) = \begin{cases} \rho_S(n_a, n_1^{s,\tau}) + \kappa(n_a, n_1^{s,\tau}, 0)\rho_S(m, n_2^{0,\tau}) & \text{if } n_a < m \\ \rho_S(n_a, n_2^0) & \text{if } n_a \geq m \end{cases} \tag{31}$$

$$\rho((s,b),\tau) = \begin{cases} \rho_S(n_a, n_1^{b,\tau}) + \kappa(n_a, n_1^{b,\tau}, 0)(p_1^{\geq m} + p_2) & \text{if } n_a < m \\ p_1^{\geq m} + p_2 & \text{if } n_a \geq m \end{cases} \tag{32}$$

$$\rho((b,b_b),\tau) = \begin{cases} \rho_S(n_a, n_1^{b_b,\tau}) + \kappa(n_a, n_1^{b_b,\tau}, 0)\left(\rho_S(m, n_3^\tau) + \delta^{n_3^\tau - m} p_1^{\geq m}\right) & \text{if } n_a < m \\ \rho_S(n_a, n_3^\tau) + \delta^{n_3^\tau - n_a} p_1^{\geq m} & \text{if } n_a \geq m \end{cases} \tag{33}$$

Proof. The statement follows by combining the actions taken under each strategy as described in Lemmas 3.1 and 3.2 with Lemma 3 by noting that the player always either subscribes, owns all the available products he plans to buy with his strategy or does not have demand.

B Proof of Proposition 4.2

Proof. From Lemma 3.1 and Lemma 3.2 it follows that, for fixed δ, γ and n_a, the normalized expected reward $w(\alpha, \tau)$ and payments $\rho(\alpha, \tau)$ for each potentially optimal strategy $\alpha \in \{b, s\} \times \{b, s, b_b\}$ are piecewise constant functions in v. As this means that the utility is differentiable and derivatives are constant, it follows that in each interval where w for all strategies is constant in v, each strategy α can at most be optimal for a single sub-interval. Starting at $v = 0$ and iteratively solving for the next point where either the optimal strategy α^* or $w(\alpha^*, \tau)$ changes thus allows us to identify intervals $I_i = [v^i, v^{i+1})$ of values in which the optimal strategy and the expected payment do not change in v.

C Data Discussion

In this section, we describe discuss the steam spy data and how it influenced our choice of distributions and parameters. We bought this dataset from the website *Steam Spy* [6], which collects publicly available data from the large video game storefront *Steam*[7] and uses it to statistically estimate the number of owners of video games over time and what percentage of them actively used the game recently (i.e., in the last two weeks). While steam employs an overly aggressive pricing strategy that partly falls outside our model and this data set is prone to estimation errors and only contains limited information about those users that bought the games, it still allows us to make a number of general observations to help find reasonable distributions. In the following, we describe the general insights we have obtained from analyzing this data.[8]

1. For most games, while the game is supported, its monthly sales stay roughly constant as long as the price of the game keeps decreasing and afterwards drops off relatively slowly. This suggests that the arrival rate of new users is roughly constant, though the quality decays, which in turn is counteracted by price drops. The exception to this is the release month (+/- 1-2 weeks), which for big releases can have $3-10$ times as many sales, as many users effectively arrived before the game's release but could not buy it yet. Additionally, the release of bigger upgrades often boosts sales of the base game, which can be explained by users in the system holding off their purchase until the upgrade releases. Denoting the overall arrivals during the first timestep by x_a, we set the arrival distribution f_a to

$$f_a(n_a) = \begin{cases} \frac{x_a}{x_a+n_{max}-1} & \text{if } n_a = 1 \\ \frac{1}{x_a+n_{max}-1} & \text{if } 1 < n_a < n_{max} \end{cases} \tag{34}$$

   While we will later vary x_a, we choose $x_a = 5$ as the standard value for most of the section.
2. While games on Steam usually do not change their base price, most games get regularly (i.e., typically every few weeks) discounted for a limited period of time, and most users buy during those discount periods. Taking this into account, the price of most games effectively drops by $40\% - 60\%$ during the first year. As the number of new owners per timestep stays roughly constant and assuming the arrival rate of users is constant, this suggests that the quality of most games for users that do buy on average decays at a rate around 10% per month, i.e., $\gamma = 0.9$. As not much more distributional information is available, for simplicity we assume for our numerics that $\gamma \in \{0.85, 0.9, 0.95\}$. We let x_γ denote the probability that γ is 0.9 and set the γ distribution f_γ to be $f_\gamma(0.9) = x_\gamma$ and $f_\gamma(0.85) = f_\gamma(0.95) = \frac{1}{2}(1-x_\gamma)$. While we will later vary x_γ, we choose $x_\gamma = 0.8$ as the standard value for most of the section.

[6] https://steamspy.com/about

[7] https://store.steampowered.com/about/

[8] The purpose of this exercise was to find *reasonable* parameter settings and distributions for the comparative statics. A detailed empirical analysis (e.g., fitting a statistical model to the data) is beyond the scope of this paper and would carry little value for this limited dataset.

3. While Steam Spy does not contain data on whether a user played a game after a certain date, it does contain data for the percentage of users who own the game and have played it in the last two week. Using this as a proxy for the percentage of users with demand $d = 1$, we see that the percentage of users who stop playing after a month for most games varies between 40% and 80% percent. For most games, once the percentage of active users has reached about 20%, it starts to only fall very slowly. Accounting for the fact that there are constantly new users buying the game, it is reasonable to assume that 20% of users have a 90% probability to not lose demand in each timestep, i.e., $\delta = 0.9$. We therefore settle on a simple two-type distribution of long-term and short-term users. Denoting the probability that a short-term user does not lose demand after one timestep by x_δ, we obtain $f_\delta(x_\delta) = 0.8$ and $f_\delta(0.9) = 0.2$. While we will later vary x_δ, we choose $x_\delta = 0.5$ as the standard value used for most of the section.
4. The dataset does not contain much information that would allow us to estimate the distribution of user values. We therefore simply set the user values to be distributed according to a normal distribution with mean $\mu = 25$ truncated to $[0, 50]$. While we will later vary the standard deviation σ of f_v, we choose $\sigma = 10$ as the standard value for most of the section.

Additionally, for the numerical analysis, we set $q_1 = 1$, $q_2 = 0.5$, $m = 6$ and $n_{max} = 12$.

D Owners, Prices and Activity Data Examples

In this section, we discuss a few representative games and provide figures to illustrate their data.

D.1 Stellaris, a 4X grand strategy strategy game

This is a so called 4X (Explore, Expand, Exploit, Exterminate) grand strategy game with regular small updates (which we do not model) and optional paid upgrades. During the observed timeframe, 4 large paid upgrades released, in October of 2016, April and September of 2017 , as well as frebruary of 2018. Each coincides with an up-tick in active players, though only the April upgrade seems to have lead to a large rise in sales. This most likely happened because it brought the quality of the whole product above a level where users that had held up on buying finally bought the base product.

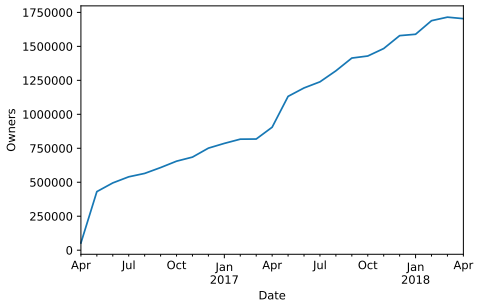

Fig. 4. Stellaris:Owners over time

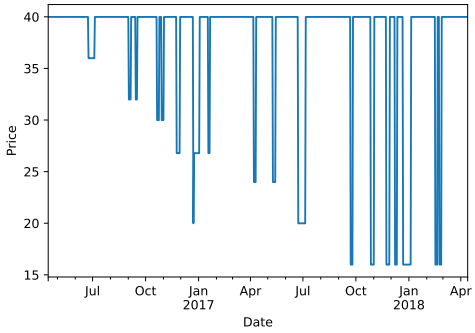

Fig. 5. Stellaris:Price over time

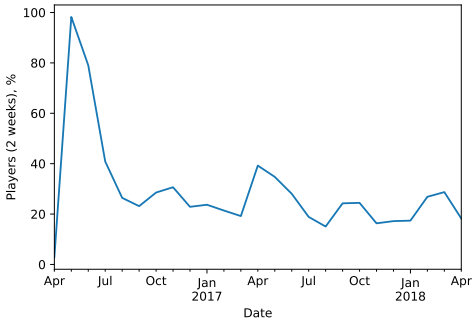

Fig. 6. Stellaris: recently active owners in %

D.2 Dark Souls III, an action RPG

This is a large, story driven game with a minor multi-player component. Famous for being very challenging. There were two paid upgrades released, one in October 2017 and one in March of 2017 that are both visible as up ticks in active players.

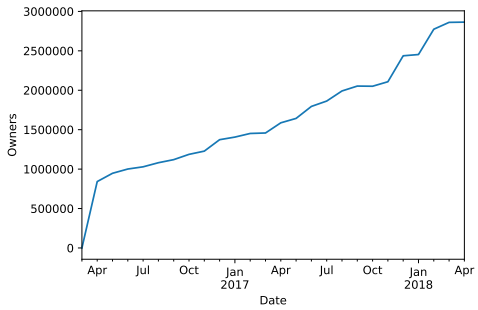

Fig. 7. Dark Souls III:Owners over time

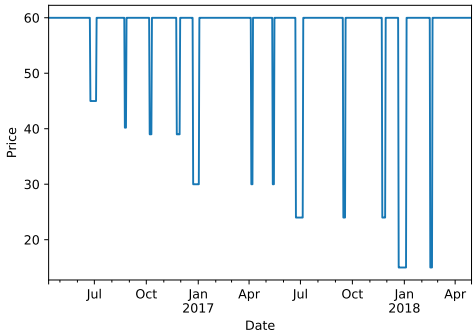

Fig. 8. Dark Souls III:Price over time

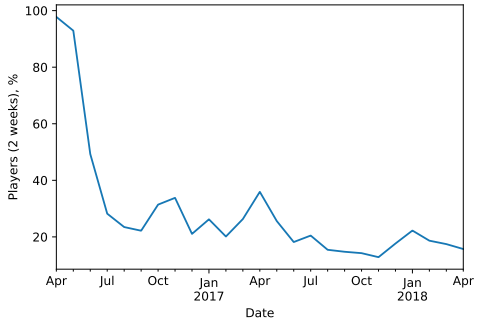

Fig. 9. Dark Souls III: recently active owners in %

D.3 Slay the Spire, a rogue-like card deck building game

This is a relatively small game that user typically play with low intensity, but for a long period of time. Additionally, in difference to the other two games, Slay the Spire had an so called 'early access' period during which an unfinished version was sold at a lower price while obtaining regular free updates. In our model, this in roughly comparable to *increasing* (instead of decreasing) the base products price when the upgrade releases, but giving out the upgrade for free.

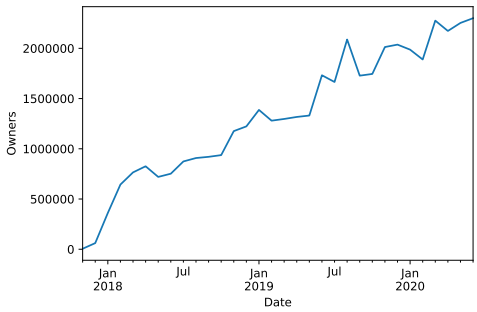

Fig. 10. Slay the Spire: Owners over time

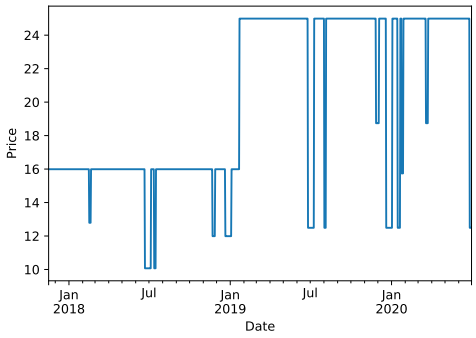

Fig. 11. Slay the Spire: Price over time

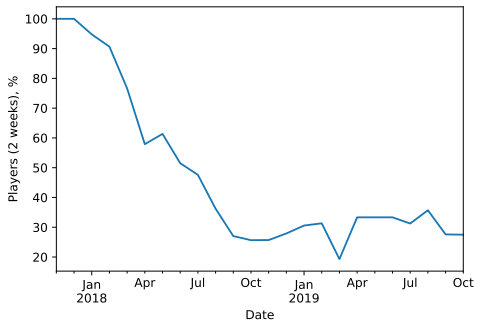

Fig. 12. Slay the Spire: Recently active owners in %

E Additional Comparative Statics

In Figure 13, we see that while increasing x_γ, and therefore decreasing the population variance of the quality decay factor γ, decreases the relative revenue potential of just

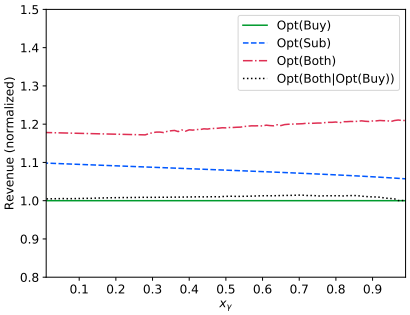

Fig. 13. Revenue for different quality decay distributions (i.e., varying x_γ)

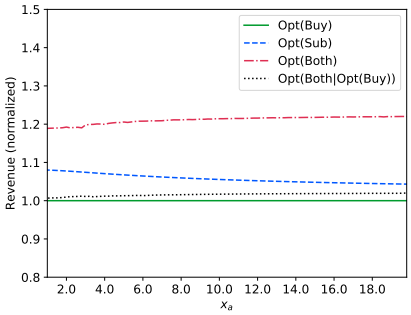

Fig. 14. Revenue for different arrival distributions (i.e., varying x_a)

offering a subscription, it can still increase the revenue potential of offering both options as market cannibalization decreases. In Figure 14, we see that how many users arrive in timestep 1 also has a relatively low impact on the relative revenue of the different strategies. Similarly to low σ, we again see that for very low x_a, the potential revenue improvement of offering a subscription option without changing the buy price goes to zero. Here this is caused by the fact that the buy price keeps decreasing because late arriving users become relatively more important for the publisher's revenue, making it harder and harder to offer a reasonably prices subscription option without loosing revenue to market cannibalization.

Curriculum Vitae

Personal Information

Name	Ludwig Dierks
Date of Birth	June 16, 1990
Place of Birth	Munich, Germany
Nationality	German

Professional Experience

June 2018 – September 2018	*Research Intern*, Microsoft Office of the Chief Economist, Redmond, Washington, USA
November 2013 – September 2014	*Student Employee*, Berner & Mattner, Munich, Germany
March 2012 – April 2012	*Intern*, iic engineers, Munich, Germany

Education

February 2016 – February 2021	*Doctoral program* at the University of Zurich, Department of Informatics, Computation and Economics Research Group
October 2012 – July 2015	*MSc in Mathematics in Operations Research* Technical University Munich, Germany
October 2009 – September 2012	*BSc in Mathematics* Technical University Munich, Germany